Mitigating bias against Non-native accents

Master thesis report

YUANYUAN ZHANG

Mitigating bias against Non-native accents

Thesis

to obtain the degree of Master of Science
in Electrical Engineering
at Delft University of Technology,
to be defended publicly on Monday May 23rd, 2022 at 9:30 AM.

by

Yuanyuan ZHANG

Faculty of Electrical Engineering, Mathematics Computer Science,
Delft University of Technology, Delft, Netherlands,
born in Heilongjiang, China.

Thesis committee:

Dr. Odette Scharenborg,	Technische Universiteit Delft
Dr. Geethu Joseph,	Technische Universiteit Delft

An electronic version of this dissertation is available at
`http://repository.tudelft.nl/`.

Contents

Abstract

Automatic Speech Recognition (ASR) systems have seen substantial improvements in the past decade; however, not for all speaker groups. Recent research shows that bias exists against different types of speech, including non-native accents, in state-of-the-art (SOTA) ASR systems. To attain inclusive speech recognition, i.e., ASR for everyone irrespective of how one speaks or the accent one has, bias mitigation is essential and necessary. In this thesis, two SOTA ASR systems (one is based on the recurrent neural network (RNN) and the other is based on the transformer architecture) are built to uncover and quantify the bias against non-native accents. Here I focus on bias mitigation against non-native accents using two different approaches: data augmentation and by using more effective training methods. For data augmentation, an autoencoder-based cross-lingual voice conversion (VC) model is used to increase the amount of non-native accented speech training data in addition to data augmentation through speed perturbation. Moreover, I investigate two training methods, i.e., fine-tuning and Domain Adversarial Training (DAT), to see whether they can utilize the available non-native accented speech data more effectively than a standard training approach. Experimental results show for the transformer-based ASR model: (1) adding VC-generated and speed-perturbed data to train the ASR model gives the best bias mitigation performance and the lowest word error rate (WER); (2) fine-tuning reduces the bias against non-native accents but at the cost of native accent performance; and (3) compared with the standard training method, DAT does not leads to further bias reduction. While for the RNN-based ASR model, all the 4 bias mitigation approaches do not show obvious benefits.

Index Terms: bias mitigation, speech recognition, data augmentation, voice conversion, domain adversarial training

Preface

Time flies and it is time to say goodbye to TU Delft. During the more than two years' studying in the Netherlands, I have lived in two cities: Delft and Eindhoven. I enjoyed the natural scenery, met many interesting people and gained the abilities to face challenges and solve problems. At my last year studying, I was glad to enter the field of ASR, be a contributor towards inclusive speech recognition and find the direction of my future work. During the master thesis project, I not only learned the knowledge in the automatic speech recognition field but also became more confident about myself.

First of all, I appreciate Dr. Odette Scharenborg, Dr. Geethu Joseph for accepting my invitation to be my thesis committee members. In addition, many thanks to my supervisor Dr. Odette Scharenborg for her patient guidance. She is helpful and warm, always letting me feel that I am supported and encouraged. Meanwhile, she has a rigorous attitude towards scientific research and takes the trouble to help me correct mistakes and clarify the logic of experiments. In my eyes, she has the image of a successful researcher and is my role model. It is worth mentioning that my daily supervisor Dr. Tanvina Patel provided me with a lot of technique supports. Every time I struggled with difficulties, she would response quickly and help me selflessly. I also sincerely thank my pre-daily supervisor Dr. Siyuan Feng. With his help and suggestions, I started this thesis project and wrote the proposal. Under Odette, Tanvina and Siyuan's guidance, I went from knowing nothing about ASR to completing this bias mitigation project and I learned how to write the paper and thesis report. Thanks for all your patience, kindness, selfless help and detailed feedback.

Second, I feel very happy and lucky to join Speech Lab Croup. In our group, everyone is kind, friendly and willing to help. I love the atmosphere and the bi-weekly meeting with all the group members. During the SALT meeting, we can know other's progress and learn from each other. Especially during the coronavirus, the SALT meeting has brought everyone together and let me know that I am not alone. Additionally, to my surprise, my classmate Yixuan Zhang and I chose the same graduation topic, allowing us to meet and discuss together, which is quite enjoyable, and I think we do more with less. Moreover, many thanks to Bence Halpern for all his help and feedback with the VC experiments and the kindly remind before each SALT meeting. Many thanks to our High Performance Computing administrator Ruud de Jong. With his help, I learned some tips to use the Linux system and ran experiments on the cluster.

Last but not least, I appreciate my parents and sister for their continuous trust and support. Specially express my deep gratitude to my boy friend Yibin Lei. After completing the undergraduate studies together, we came to the Netherlands. Here we obtained diverse knowlegde and experienced colorful life together. Thanks to Yibin, I have never felt lonely. Furthermore, I gained a lot of friendship not only in TU Delft, but also in TU Eindhoven. Sincere thanks to all my friends for your concern and support over the past two years. All the common experience will be left in my mind and became the most beautiful memories.

All in all, living and studying in the Netherlands has been a great experience in my life! Thanks for myself to insist studying, find such an interesting thesis topic, join such a nice group and complete this project. Hope everyone I met during this trip can have a better future! Will miss and appreciate you forever.

Yuanyuan Zhang
Eindhoven, Apr. 2022

1

Introduction

1.1 Motivation

Automatic Speech Recognition (ASR) has improved a lot since the introduction of deep learning techniques [1–8]. We can see ASR applications everywhere making people's life more convenient, such as the voice assistants on the smart phone (Apple's Siri, Google's Alexa, etc.) and the voice interaction systems in hospitals, banks, governments and telephone customer services.

Currently, state-of-the-art (SOTA) ASR systems work extremely well for speakers whose speech patterns match its training data: typically, these are adult highly-educated first-language speakers of a standardized dialect, with little or no speech disability (referred to as norm speakers). Anecdotal and recent empirical evidence, however, have shown that for many groups of people ASR works less well [9][10], even when the ASR systems are trained on the speech of that speaker group [11]. In other words, SOTA ASR systems are biased against speakers whose speech deviates from norm speakers. In order to allow minority groups to use ASR on an equal footing with norm speakers and let ASR technology help more people, i.e., to achieve inclusive ASR, recently, the authors of [9] uncovered and quantified bias regarding to gender, age, regional accents and non-native accents in a SOTA Dutch hybrid ASR system for both read speech and human-machine interaction (HMI) dialogue speech.

According to the experimental results in [9], among the factors of gender, age, regional accents and non-native accents, the bias against non-native accents is the biggest. Meanwhile, the global migration has increased in the past few decades [12]. Across Europe, most of those surveys said their countries become more diverse in the past two decades including the Netherlands [12]. In order to promote the integration of immigrants and international students into the local life better, mitigating the bias against non-native accents in SOTA ASR systems is quite important.

There is very limited research focusing on bias mitigation against non-native accents, and most existing research focuses on accented speech recognition. These studies [13–18] aim to improve the performance of non-native accented speech recognition, i.e., lowering the word error rate (WER). However, in order to build inclusive ASR for treating non-native speakers the same way as native speakers, i.e., ASR for everyone irrespective of how one speaks or the accent one has, the aim should not be just to lower the WER, but also to pay close attention to and reduce the performance gap between non-accented and accented speech. This performance gap between accented and non-accented speech is normally referred to as "bias". And previous accented speech recognition works do not concern such gap. For example, in [14], the WER on non-native accented speech improved, while at the same time the WER on native accented speech improved more. As a result, the bias against the non-native accented speech increased, i.e., the gap between the native and non-native speakers has grown. As such the objective of this thesis is to explore the methods for mitigating the bias against non-native accents. I concern the WERs of native and non-native accented speech data, and the gap "bias" between them. The ideal result is that the "bias" becomes smaller and the WERs become lower as well.

The mismatch between the training data and the test data may be one possible reason that causes the bias against non-native accents. For instance, in [9], the authors only used the standard Dutch speech data as the training data to train ASR systems but tested with both native and non-native accented speech data. Their experimental results showed that such ASR systems existed big bias against non-native accents. For the same Dutch words, the native and the non-native accented Dutch speakers may have different articulation due to the different speaking styles. Consequently, adding non-native accented speech data to the training data set is a possible solution for non-nativeness bias mitigation. However, there is only very little non-native accented Dutch training data available. Actually, even for the most widely used languages like English, compared with the norm accented speech data set, the sizes of non-native accented speech data sets are relatively smaller [15] [14] [16] [13] [18] [19]. Furthermore, even though Dutch is a popular language, there are far fewer Dutch speakers than English speakers. Thus it is quite difficult to collect non-native accented Dutch speech data. In addition, for the Dutch accented speech data researched in this thesis, high variability also appears inside the non-native accented speech data set (the data set contains only very few speech data but with non-native speakers coming from about 37 different birth countries), which makes mitigating the non-nativeness bias more difficult and challenging.

Considering the lack of non-native accented Dutch speech data, the non-native accented Dutch speech recognition is typically a low-resource problem. Data augmentation techniques have been widely used to increase the amount of the training speech data [9] [19]. Common data augmentation techniques include speed perturbation [20] (this traditional data augmentation method will be explored further), spectrogram augmentation [21], adding noise [22] and reverberation [23]. Specifically, for the speed perturbation method, it produces a warped time signal, resulting in a change in the duration of the audio signal [20]. For the spectrogram augmentation method, it directly uses different modification methods to change the features of the speech data including warping the spectrogram in the time direction, masking blocks of consecutive frequency channels, and masking blocks of utterances in time [21]. Corrupting the clean training speech data with various additive and convolutive noises is helpful to increase the noise robustness of ASR models [19]. Reverberation is typically represented by convolution of the audio signals with a room impulse response and it can be used to generate multi-conditional speech data [23]. All of these data augmentations either make adjustments directly to the speech signal or the spectrogram of the speech signal. Even though the past experimental results have shown these data augmentation methods are able to improve the ASR performance, they did not create the “new” speech data like the speech data spoken by new speakers. To increase the diversity in the training data set and obtain "new" speech data spoken by unseen speakers, voice conversion (VC) that generates speech data with the characteristics of new speakers (i.e., new voices) but keeps the linguistic information can be a possible solution. The VC-generated speech data has the potential to improve ASR performance, like in [19], voice transformation based data augmentation is proposed for improving the performance of foreign accented speech recognition. The voice transformation technique is capable of manipulating the vocal-source and vocal-tract characteristics to alter the speaker’s voice quality and/or impart novel speaker identities [19]. In recent years, the VC techniques based on deep learning has shown substantial development and are able to change the speaker identities of the speech data well [24–26]. In [25], an autoencoder-based VC model trained on data from LibriSpeech [27] and Libri-Light [28] (English data sets) was respectively applied on speech data of four unseen languages (Afrikaans, Setswana, isiXhosa and Sepedi) for the purpose of data augmentation. For each language, the augmented speech data successfully improved the ASR performance in very low-resource settings (roughly 10 minutes original training data). VC thus has the potential to augment non-native accented speech data, improve non-native accented speech recognition, and then narrow the gap between native speakers and non-native speakers in terms of ASR performance. Based on the fact that we only have few non-native accented speech data, I would like to generate more non-native accented speech data with the voices from unseen speakers by utilizing the VC technique cross-lingually. To be more specific, the VC model is trained with a combination of the non-native accented Dutch speech data and English speech data. While generating new non-native accented speech, I take the Dutch speech as the source speech and the characteristics of English speakers as the target voice identities.

In addition to the straightforward solution: increasing the amount of the non-native accented speech data, we can also get inspired from previous accented speech recognition works. In order to make ASR systems more robust to non-native accents, various training methods have been proposed including contrastive learning [13], domain adversarial training (DAT) [14] [16] [18], multi-task learning [17] and transfer learning [14]. Specifically, both contrastive learning and DAT are domain adaption methods, which can be used to minimize the difference between the standard speech data and the non-native accented speech data. However, applying contrastive learning for accented speech recognition needs parallel data which we do not have (the parallel data means that both the norm speakers and the non-native speakers speak the same linguistic contents). [16] proves that performing gradient reversal in domain adversarial training (DAT) is equivalent to minimizing the difference of output distributions of different accents. DAT [29] improved the performance of accented speech recognition for both end-to-end (E2E) [18] and hybrid ASR systems [29]. The experimental results in [29] showed that the performance of DAT was better than that of multi-task learning. Combining DAT with transfer learning further improved the performance of the accented speech recognition [15]. Since DAT and transfer learning have shown benefits for building non-native accents robust English ASR systems, it is worth exploring if DAT and fine tuning (one kind of transfer learning) are able to mitigate the bias against non-native Dutch accents.

1.2 Research questions

The objective of this project is to mitigate the bias between native and non-native accents for E2E SOTA ASR systems, which means bridging the gap between native speech recognition and non-native speech recognition without hurting the recognition performance of native speech. Currently, the E2E ASR has achieved lower WERs than the conventional hybrid approaches, as shown in the website PapersWithCode[1]. Since the bias against non-native accents has been uncovered and quantified in [9] only for hybrid ASR systems, I would like to quantify the bias against non-native accents and mitigate such bias for the SOTA E2E ASR systems i.e., recurrent neural network (RNN)-based and transformer-based ASR systems. Hence the main research question of this thesis is:

- How to mitigate bias against non-native accents?

I consider the main research question from two angles: increasing the amount of the non-native accented speech data in the training data set by data augmentation methods i.e., speed perturbation and cross-lingual VC; and realizing domain adaption by applying training strategies i.e., fine-tuning and DAT. Thus the main research question can be divided into two sub-questions:

- **RQ1:** Are non-native speech data augmentation methods (i.e., speed perturbation, VC-based data augmentation) helpful for bias mitigation against non-native accents in the two SOTA E2E ASR models (i.e., RNN-based and transformer-based ASR models)?

- **RQ2:** Do the DAT or fine-tuning have the ability to use the available non-native accented speech data more effectively than the standard training method, resulting in bias reduction?

I conduct extensive experiments to find the answers of the research questions. First, I use the SpeechBrain [30] toolkit to build two SOTA Dutch ASR models and quantify the bias against non-native accents respectively. As there are quite few research on Dutch E2E ASR system, I choose the two SOTA ASR models according to models' performance on English datasets. Afterwards, for both models, I conduct the data augmentation and the training strategies experiments to find answers corresponding to RQ1 and RQ2. Specifically, I work on Dutch ASR with read and conversational types of speech:

- **Read speech:** speakers read the given sentences from newspapers, books, papers and so on.

- **Conversational speech:** it is recorded during the conversation such as human-to-human interaction and human-to-machine interaction. It is a complex behavior to force speakers to satisfy the social demands connected with the spoken language [31]. In the ASR field, the conversational speech is always more difficult to recognize than read speech [9] [31] [32].

[1] `LinktoPapersWithCode`

1.3 Outline

This thesis composes of several chapters. In Chapter 2, the necessary background knowledge is given for better understanding of this thesis; In Chapter 3, the methods and the corresponding experiments designed for exploring the research questions are illustrated; In chapter 4, the experimental results are described. Based on the experimental results, discussions, conclusions and the future work are provided in Chapter 5.

2

Background

In this chapter, I provide the required knowledge of this thesis and present the relevant works of bias research in ASR. I start with ASR related knowledge in Section 2.1 i.e., traditional ASR, deep learning techniques and E2E ASR. Next, I introduce the SOTA non-parallel VC model AGAIN-VC and the speed perturbation technique I used in the following experiments in Section 2.2. Besides, I give a brief introduction of the two training strategies used in this thesis i.e., fine-tuning and domain adversarial training in Section 2.3. Moreover, the evaluation metrics used in this thesis are introduced in Section 2.4. Finally, I present the relevant bias works in Section 2.5.

2.1 ASR

ASR is a modeling task that processes human speech into the corresponding text sequence after recognizing. To be more specific, given a speech signal $\mathbf{X}$ and corresponding transcriptions $\mathbf{Y} = (y_1, y_2, ..., y_L)$, where the elements in $\mathbf{Y}$ are words in the sequence, an ASR system aims to learn how to model the distribution $P(\mathbf{Y}|\mathbf{X})$. Given a new speech signal $\mathbf{X}^*$, the predicted transcription $\mathbf{Y}^*$ is obtained by $\mathbf{Y}^* = \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}^*)$.

2.1.1 Traditional ASR

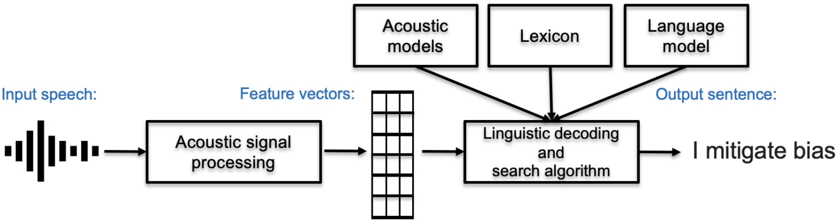

Figure 2.1: The traditional hybrid ASR system.

The traditional hybrid method has dominated the ASR field before the emergence of the deep learning techniques [33]. The architecture of the traditional hybrid ASR system is shown in Figure 2.1, consisting of multiple separate models: the acoustic model, the lexicon model and the language model. In Figure 2.1, the input speech data is first preprocessed into acoustic feature vectors via signal processing methods, and each feature vector represents the information of the speech signal within a short time. The speech data uploaded to the computers is not the continuous signal anymore, but the digital signal. For instance, if the sample rate is 16000 Hz, there will be 16000 numbers for 1 second speech data, which is too large. Thus, a feature extraction process to make the speech data more compact is needed.. Using mel-spectorgram features to represent the speech data is quite common [24], and in this project, I use the mel-spectrogram as the input features as well. The spectrogram is a 2-dimensional representation method for the speech data, with the time information in the x-axis and the frequency information in the y-axis. The mel-spectorgam is also a spectrogram where the frequencies in the y-axis are converted into mel scale. The mel bin represents the resolution on the y-axis.

Afterward, the feature vectors are forwarded to the hybrid ASR model. The goal of the hybrid ASR model is to find the most likely output sentence $\mathbf{Y}$, given the acoustic feature vectors denoted by $\mathbf{X}$. One approach is to look for all possible sequences of words with fixed maximum length and finds the one matching the input feature vectors most, described as the Equation 2.1.

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} P(\mathbf{Y}|\mathbf{X}) \tag{2.1}$$

According to the Bayes's theorem, Equation 2.1 can be reformulated into Equation 2.2.

$$\mathbf{Y}^* = \arg\max_{\mathbf{Y}} P(\mathbf{X}|\mathbf{Y})P(\mathbf{Y}) \tag{2.2}$$

where $P(\mathbf{X}|\mathbf{Y})$ corresponds to the acoustic model in the Figure 2.1. The acoustic model aims to model how likely a sequence of phones (the speech of each word consists of multiple phones) appears given a sequence of feature vectors. The lexicon model in Figure 2.1 is a model that maps phones into words. And the language model in Figure 2.1 models the likelihood of the word sequence, corresponding to $P(\mathbf{Y})$ in the Equation 2.2, predicting the probability of a sequence of phones $\mathbf{Y}$ occurring in a language. All of these 3 models need to be trained individually.

2.1.2 Deep Learning for ASR

Deep Learning basics

Deep learning is a class of machine learning techniques that exploit many layers of non-linear transformation operations, for solving pattern analysis and classification problems [34], [35], [36]. Deep learning algorithms are built upon neural network layers, each of which consists of small individual units named neurons which perform non-linear transformations.

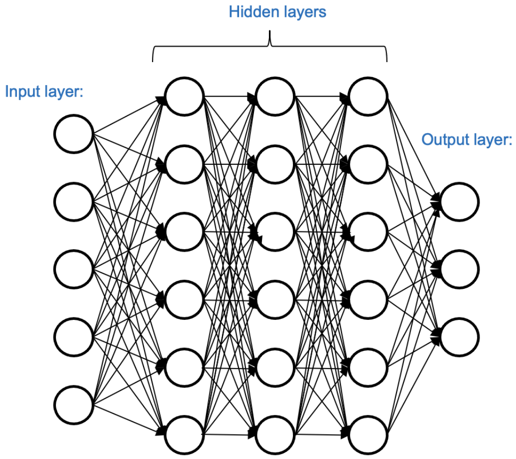

Figure 2.2: A FNN example.

The feedforward neural network (FNN) is the simplest neural network and the basic unit for composing complex deep models, also named as multilayer perceptron. The information in such network can only move in one direction: forward. An example FFN is shown in Figure 2.2, where the input layer stores the inputs like speech signals, hidden layers process the input data and the output layer outputs the desired outputs like text transcriptions. Given the input data e.g., speech, and the output data e.g., human-annotated speech transcriptions, the FFN aims to learn a mapping function f^* that approximates the "optimal" mapping f that maps the input data X into the output data Y with 100% accuracy, i.e, for each corresponding $x \in X$ and $y \in Y, f(x) = y$. The learned function f^* is determined by the structure of the network and the parameters θ inside the network.

To obtain a neural network that can conduct classification tasks as accurate as possible, the learned f^* should be as close as f. The objective/loss function is a function that measures how far the predicted outputs of the network are from the desired/true outputs, indicating how well f^* approximates f. As such the training/learning process of a neural network is indeed an optimization process, i.e., a process of finding parameters θ that minimize the objective function. Different tasks (e.g., ASR and VC) have different corresponding objective functions. For ASR, the CTC loss can be used and for VC, the L1 loss can be used (both the CTC loss and the L1 loss will be introduced afterwards). Gradient Descent is the most common optimization technique for learning deep neural networks. To minimize the objective function, the Gradient Descent iteratively updates the parameters with the guidance from the gradient of the training samples. To be more specific, for each time, the objective function gradient with respect to the parameters θ is computed and then θ is updated through the opposite direction of the gradient to minimize the value of the objective function. Some popular Gradient Descent methods are Stochastic Gradient Descent (SGD), Momentum and Adaptive Moment Estimation (Adam).

The sequence to sequence model composed of the encoder and the decoder is first introduced by Google in [37]: a sequence to sequence model aims to map a fixed-length input with a fixed-length output where the length of the input and output may differ. Encoder is an architecture that converts a sequence into a single vector and decoder is another architecture that is able to convert the coded message/single vector back to a sequence. Speech recognition is a use-case of the sequence to sequence model. Specifically, the speech data is the input sequence while the corresponding transcriptions are the output sequence. The recurrent neural network (RNN) and transformer are suitable for processing the sequence data.

RNN is a typical and widely-used neural network for processing sequential data e.g., speech data. Different from FFN, the structure of RNN is specifically designed for exploiting the sequential information of the data. This property is essential and useful for dealing with data like speech, as knowing what previous words has been spoken is quite useful for recognizing current spoken words. As shown in Figure 2.3, the output of the hidden layers flows not only to the output layer but also to the hidden layers itself. With such loop the previous sequential information can iteratively be stored in the hidden layers and the hidden layers can be seen as a short-term memory unit. The gradient vanishing and exploding issue is a major problem with the RNN. To overcome the shortage, various RNN variants were proposed, e.g, Gated Recurrent Unit (GRU) [38].

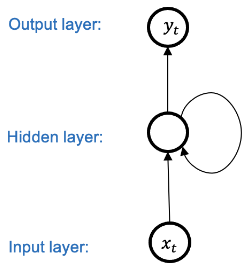

Figure 2.3: A RNN example.

Transformer is the SOTA neural network architecture which has been successfully applied in sequence to sequence task, with the advantage of fast iteration speed in the training stage, because compared with RNN, the transformer has no sequential operations [39]. In [40], experiments revealed various training tips and significant performance benefits obtained by using transformer including the surprising superiority of transformer in 13/15 ASR benchmarks in comparison with RNN. Furthermore, now the SOTA ASR systems are mostly built upon the transformer [1–8].

2.1.3 E2E ASR

The E2E ASR aims to directly map a sequence of feature vectors e.g. mel-spectrogram into a sequence of words. Compared with the traditional hybrid ASR model, the E2E ASR model is easier to train as the training process is integrated. However, the E2E ASR models are more data-hungry. With few speech data, the hybrid ASR is able to achieve better performance than the E2E ASR.

CTC Loss

The E2E ASR can be treated as a modeling task, to be more specific, given processed spectrogram of the audio denoted by $X = (x_1, x_2, ..., x_T)$ and corresponding transcriptions denoted by $Y = (y_1, y_2, ..., y_L)$, where $T >= L$ and the items in Y belong to distinct labels. In the E2E ASR, the labels are generated by tokenizers. The categories of the labels in the E2E ASR includes characters, words, sub-words.

Connectionist temporal classification (CTC) [41] is an loss function that allows RNNs/-transformers to be trained for sequence transcription tasks without requiring any prior alignment between the input and target sequences. It uses the intermediate label representation denoted by $\pi = (\pi_1, \pi_2, ..., \pi_T)$, allowing repetitions of labels and occurrences of a blank label (-), which represents the special emission without labels. CTC trains the model to maximize $P(Y|X)$ as denoted by equation 2.3:

$$P(Y \mid X) = \sum_{\pi \in \Phi(Y')} P(\pi \mid X) \tag{2.3}$$

where $\pi \in \Phi\left(Y'\right)$ presents the probability distribution over all possible label sequences and Y' is a modified label sequence of Y, which is made by inserting the blank symbols between each label and the beginning and the end for allowing blanks in the output (i.e., Y = (c, a, t),Y' = (-, c, -, a, -, t, -)). CTC is generally on top of the RNNs/transformers. Here I take the RNN architecture as an example. Each RNN output unit is interpreted as the probability of observing a corresponding label at particular time. The probability of label sequence $P(\pi|X)$ is modeled as being conditionally independent by the product of the network outputs as the equation 2.4.

$$P(\pi \mid X) \approx \prod_{t=1}^{T} P\left(\pi_t \mid X\right) = \prod_{t=1}^{T} q_t\left(\pi_t\right) \tag{2.4}$$

where $q_t(\pi_t)$ denotes the softmax activation of π_t intermediate label in RNN output layer q at time t. The CTC loss is then defined as the negative log likelihood of the ground truth transcriptions Y^* as the equation 2.5.

$$\mathcal{L}_{\text{CTC}} \triangleq -\log P\left(Y^* \mid X\right) \tag{2.5}$$

Transformer-based joint CTC/attention E2E ASR model

Even though compared with RNN, transformer has faster iteration speed, it takes more epochs for transformer to converge. Joint transformer-based CTC/attention training and decoding for ASR is proposed to speed up convergence [39]. The CTC joint decoding is implemented by adding a new forwarding branch from the transformer encoder.

First, we could have a look on the transformer-based E2E ASR system as an example. The input speech is first represented as a sequence of 80-dimensional mel-spectrogram features denoted by $X \in \mathbb{R}^{T\times 80}$, T is the input length. First, we subsample the X into $X^{sub} \in \mathbb{R}^{n_{sub}\times d^{att}}$ by one-layer CNN. Next, the transformer encoder encode X^{sub} into $X_e \in \mathbb{R}^{n_{sub}\times d^{att}}$ and feed it into the following transformer decoder. Transformer decoder predicts all the tokenizer frames as $P(Y^*|X_e)$. The transformer-based sequence to sequence ASR model loss is illustrated by equation 2.6.

$$\mathcal{L}_{\text{s2s}} \triangleq -\log P(Y^*|X_e) \tag{2.6}$$

When join training the CTC and transformer based sequence to sequence model, the output of the transformer encoder X_e will be also input to the CTC loss in equation 2.5, so the CTC loss now is as equation 2.7. On this occasion, the joint CTC/attention loss is denoted by equation 2.8 [39].

$$\mathcal{L}_{\text{CTC}} \triangleq -\log P(Y^*|X_e) \tag{2.7}$$

$$\mathcal{L}_{\text{joint}} = -\alpha\mathcal{L}_{\text{s2s}} - (1-\alpha)\mathcal{L}_{\text{CTC}} \tag{2.8}$$

Where $\alpha \in \mathbb{R}$ is a hyperparameter to control the degree of influence of the CTC loss on the whole model.

2.2 Data augmentation for speech data

2.2.1 Voice conversion model: AGAIN-VC

Voice conversion is a technique to convert the voice of a source speaker's speech to that of a target speaker's speech while maintaining the same linguistic content [24]. The source speech data is the data you would like to keep the content but change the voice. The target data is the data you would like to ignore the linguistic content but obtain the voice. Here I introduce a SOTA non-parallel autoencoder-based VC model: AGAIN-VC [24]. The autoencoder is normally composed of an encoder and a decoder. The encoder is used to compress the input feature vectors and the decoder aims to reconstruct the input features. "Non-parallel" means that the AGAIN-VC model does not need to be trained by parallel data. In the VC field, parallel data means that both the source and the target speech are with the same linguistic content.

The main idea of AGAIN-VC is to use the single encoder to extract both the speaker information and the content information. When a certrain speaker speaks, his/her voice characteristics keep unchanged while the linguistic contents are varied with the time. Hence AGAIN-VC treats the speaker information as the global style of a speech, which is supposed to be time-invariant. On the other hand, the content information is supposed to be time-varying.

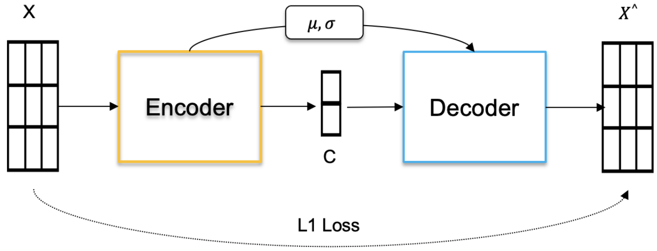

Figure 2.4: AGAIN-VC model architecture.

The main structure of AGAIN-VC model is illustrated in Figure 2.4. The input data of the AGAIN-VC is not the raw speech, but the mel-spectrogram. During the training stage, only one piece of the speech data is input to the AGAIN-VC model. First the mel-spectrogram **X** will be input to the encoder. Now **X** contains both the voice and the content information together. The output of the encoder will be treated as the content information denoted by **C**, while the time-invariant channel-wise mean μ and the channel-wise standard deviation σ calculated by instance normalization (IN) layers contained in the encoder are treated as the voice information. Specifically, the IN layers are used to disentangle the speaker information i.e., μ, σ from the input mel-spectrogram. Next, the speaker information μ and σ calculated by the IN layers and the output of the encoder denoted by **C** are combined through the adaptive instance normaliztion layers (AdaIN) contained in the decoder, leading to the reconstructed output mel-spectrogram denoted as $\hat{\mathbf{X}}$. Then the generated mel-spectrogram $\hat{X}$ will be converted in to waveform by a pre-trained MelGAN vocoder [24]. L1 Loss in figure 2.4 denotes the self-reconstruction loss used during training stage, as illustrated in Equation 2.9.

$$\mathcal{L} = \|X - \hat{X}\|_1^1 \tag{2.9}$$

For the inference stage, suppose the mel-spectrogram of the source speech data (the data you would like to keep the content but change the voice) is denoted as **S**, while that of the target speech data (The target voice you would like to convert to) is denoted as **T**. Then the voice conversion can be reached by mainly two steps:

1. Extract the speaker information of **T**, including $\mu(\mathbf{T})$ and $\sigma(\mathbf{T})$,

2. Pass the $\mu(\mathbf{T})$, $\sigma(\mathbf{T})$ and **S** to the decoder. With the AdaIN layer, the voice conversion from the target speaker to the source speaker is achieved, as Equation 2.10.

$$\mathrm{AdaIN}(\mathbf{S}, \mu(\mathbf{T}), \sigma(\mathbf{T})) = \sigma(\mathbf{T})\mathbf{C}(\mathbf{S}) + \mu(\mathbf{T}) \tag{2.10}$$

Where the **C**(**S**) represents the content information of the source speech data.

2.2.2 Speed perturbation

Speed perturbation is widely used for speech data augmentation, producing warped time signals [20]. With speed perturbation, the duration of the speech data is changed. The size of the perturbation factor reflects whether accelerating or decelerating the speech signal. When the value of perturbation factor is bigger than 1, then the data will be accelerated. When the value of perturbation factor is smaller than 1, then the data will be decelerated.

2.3 Training strategies

In addition to the standard training method i.e., training the neural networks with only data of the target task from scratch i.e., a randomly initialized model, fine-tuning and DAT are also investigated in this thesis.

2.3.1 Fine tuning

Transfer learning aims to take the knowledge from another tasks to solve a new but related problem. Fine-tuning is a kind of transfer learning methods where a neural network trained starting from a pre-trained model. With the fine-tuning method, what the pre-trained model has already learned helps with the other task without having to learn it from scratch.

2.3.2 Domain Adversarial training (DAT)

Domain Adversarial Training (DAT) is a popular and common training strategy for domain adaptation [18, 42]. The main assumption of DAT is that the training and test data come from different distributions such that an effective domain transfer from the training/source domain to the test/target domain is needed. DAT suggests that to achieve successful domain transfer, the extracted features should be domain-invariant i.e, the features should not be able to discriminate between the training and test domains such that the classifier learned with the source domain can be directly applied to the target domain. Thus DAT aims to obtain features that are not only discriminative to the targeting task but also invariant to the shift of the data domain.

To learn such features, as shown in Figure 2.5, two discriminative classifiers are applied and jointly optimized with the feature extractor: (1) a label predictor predicting the class labels for the targeting task e.g., ASR for our case and is used while both training and testing; and (2) a domain classifier that predicts which domain (source or target) the extracted features belong to and is used only during training. During training, the parameters of the two classifiers are optimized to minimize the corresponding error on the training samples, and the parameters of the feature extractors are optimized to minimize the error of the label predictor but maximize the error of the domain classifier. The latter optimization direction is opposite to the optimization direction of the parameters of the domain classifier and encourages learning domain-invariant features.

Domain Adversarial Neural Networks (DANN) is the neural network that incorporates DAT strategy [42, 43]. The DANN architecture is shown in Figure 2.5.

First, the feature extractor extracts valuable features from the input X and then feed the features to the domain classifier and class predictor. Owing to the chain rule, the gradient of the parameters of the feature extractor can directly effects the gradient of the parameters of the domain classifier and label predictor and thus influence the performance of the label predictor and the domain classifier.

Second, the domain classifier predicts whether the extracted features belong to the source or the target domain and is optimized to discriminate the features as accurate as

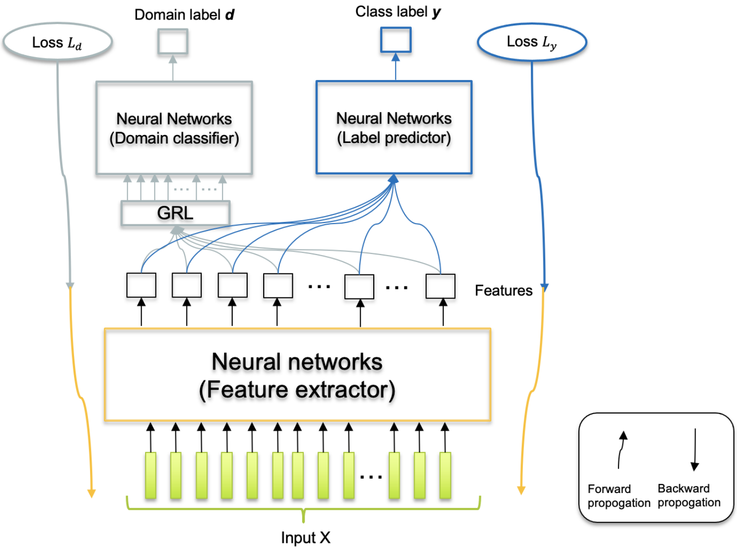

Figure 2.5: The DANN architecture incorporating DAT strategy.

possible. However, the feature extractor is adversarially optimized to fool the domain classifier as much as possible, which means the extracted features are forced to be domain-invariant i.e., indistinguishable between source and target domains. To achieve this, the Gradient Reversal Layer (GRL) is the key component, which is positioned between the domain classifier and the feature extractor. The GRL reverses the gradient of the loss of the domain classifier and transit the reversed gradient to the feature extractor. As such the reversed gradient forces the feature extractor to extract domain-invariant features. The GRL layer only works during gradient descent and has no effect on forward process.

Last, the label predictor predicts the task labels e.g., text sequences in ASR, and is optimized to make as fewer mistakes as possible. The gradient of the label predictor is also transmitted to the feature extractor. To conclude, the two main objectives of DANN are correctly predicting class labels and reducing the difference between features in the source and target domains.

2.4 Evaluation metrics

2.4.1 WER

WER is a common and widely used metric to evaluate the performance of ASR systems. It is calculated according to the equation 2.11. The lower the WER is, the better the ASR system is.

$$\mathrm{WER} = \frac{(\mathrm{S} + \mathrm{I} + \mathrm{D})}{\mathrm{N}} * 100\% \tag{2.11}$$

Where:

- S is the number of word-level substitutions.

- I is the number of word-level insertions.
- D is the number of word-level deletions.
- N is the number of words of the reference text, i.e, the correct text sequence of the speech.

For instance, for a speech with reference “Hello there”, recognizing the speech with transcription “Hello bear” will obtain a WER of 50%, as there exists one word-level substitution (“here” to "bear“) and the reference has two words. Moreover, recognizing the speech with ”Hello“ also gets a WER of 50% since a deletion appears. Recognizing the speech with ”Hello everyone here“ will lead to a WER of 50% as well since "everyone" is inserted.

2.4.2 Bias quantification

In this thesis, the bias against non-native accents is defined as the WER gap between the native speech accented data and the non-native accented speech data tested on the same ASR model.

2.5 Related works on bias research

Since ASR appears to be more popular and frequently-used in people’s lives, it is essential and critical to make sure the ASR systems deployed in real-life can deal with the high variability in the human speech e.g., variability caused by the difference between genders, ages and regions equally well. However, despite the great success of deep learning on ASR [44–46], recent practice and studies suggest that SOTA ASR systems are not able to recognize the speech of all groups of people equally well. The performance gap between different groups of people can reflect the biases existing in the society e.g., gender, race, and age biases.

Multiple studies have shown that there exists a performance gap between male and female speakers in Arabic [47], French [48] and English [48, 49]. [47] found bias exists not only in genders but also in ages: they found the ASR system recognized speakers younger than 30 years old better than the speakers older than 30 years old. A clear performance gap between black and white speakers was found in [50] as well. In addition, child speech recognition is also proven to be more difficult than adult speech [51]. Another important challenge for current ASR systems is the speech impairment, i.e., ASR systems perform quite bad on speech spoken by people with dysarthria [52], oral cancer [53], and cleft lip and palate [54]. Last but not least, speech variance in accents and regions has also shown effecting the ASR performance a lot [55, 56]. However, most of the above works only focus on one or two factors that contributes to the degradation of ASR performance, a systematic and comprehensive bias evaluation for ASR models is needed. To this end, [57] systematically uncovered the bias in ASR systems by investigating the Dutch ASR performance on speech data spoken by groups of people with different genders, ages, nationalities and regions. There results showed the biases existing in the society are perpetuated in current SOTA ASR systems and remind us the great importance to make ASR inclusive and reliable.

3

Methodology

In this chapter, the databases and bias mitigation methods used in this thesis are introduced in detail. More specifically, in Section 3.1, a through description and explanation of the chosen databases are given, e.g, the reason of the database choices, the training and test sets' settings, etc. Moreover, the designed 2-step methodologies (shown in Figure 3.1) and the set-ups of the experiments conducted for answering the research questions are presented. First, the implementation of the 2 SOTA ASR baselines and bias quantification are introduced in Section 3.2. Second, the implementation of the bias mitigation methods i.e., non-native accented speech data augmentation and effective training strategies are respectively given in Section 3.3 and Section 3.4.

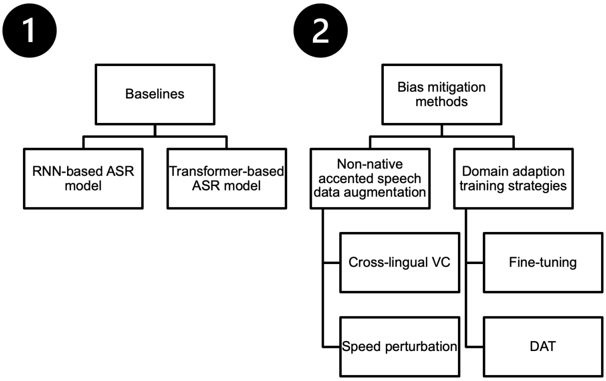
1
2
Baselines
RNN-based ASR model
Transformer-based ASR model
Bias mitigation methods
Non-native accented speech data augmentation
Domain adaption training strategies
Cross-lingual VC
Speed perturbation
Fine-tuning
DAT

Figure 3.1: The overview of the 2-step methodologies.

3.1 Datasets

In this section, all the data sets used in this thesis are introduced. The data sets mainly have two training applications i.e., training ASR models and VC models. Both the spoken Dutch corpus (CGN) (see Section 3.1.1) and the Jasmin-CGN corpus (see Section 3.1.2) are used for ASR training. Furthermore, the jasmin-CGN corpus and the VCTK corpus (see Section 3.1.3) are used for VC experiments.

3.1.1 The spoken Dutch corpus (CGN)

The CGN [58] is a Dutch corpus containing native speech data spoken by speakers from the Netherlands and Flanders. In this thesis, I concern the bias against non-native accents but not regional accents between the Netherlands and Flanders, so I only use the data recorded in the Netherlands to train our E2E ASR systems. When mentioning CGN later, I refer to the speech data recorded in the Netherlands.

The CGN corpus consists of monologue and multilogue speech data spoken by speakers with the 18 - 65 age range. It has 15 different speech data components listed below with brief descriptions. 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.

In this thesis, the CGN training and test sets partitions follow the experimental set-up in [9]. More specifically, the ASR model built in [9] is implemented based on an open-source github repository[1] containing detailed training, test sets partitions and pre-processing procedures. The unprocessed training data consists of the speech data components *a* - *d* and *f* - *o*, with 483-hour materials in total, spoken by 1185 female and 1678 male speakers. The github repository do not use the component *e* for training but do not give the reason. To pre-process the CGN training data, they first segment the long recordings into small chunks and then remove the silence parts, resulting in 423 hours processed training data. To obtain a more proper (bigger) training batch size, I ignore relatively long audios (audios with more than 20s duration), as the goal of the thesis is to explore the possible solutions for non-native accents bias mitigation instead of achieving the lowest WER. This ignoring procedure finally yields 380.12 hours of standard Dutch speech data, denoted by $\mathbf{C_{train}}$. Following the test data list[2] given by the github repository, I make 2 test sets i.e., the broadcast news (BN) test set denoted by $\mathbf{C_{BN}}$ and the conversational telephone speech (CTS) test set denoted by $\mathbf{C_{CTS}}$. Even though the name of the test data list in the github repository is "nbest-dev-2008" and authors of [9] claim that their test sets set-up follows the experimental set-up of the paper "Results of the N-Best 2008 Dutch Speech Recognition Evaluation [59]", the test sets they and I used are similar but not exactly same with the test sets used in paper [59]. To be more specific, compared with [59], we use the test sets with the same categories but with smaller sizes. There are no reasons given for choosing BN and CTS data to build the test sets in the github repository and papers [9] and [59]. From my observation of the data components in CGN, I find that the BN speech data and CTS speech data are respectively corresponding the monologue and multilogue speech data in CGN. Selecting BN and CTS speech data for test is able to show the speech recognition performance on the CGN corpus.

The $\mathbf{C_{BN}}$ comes from the component *k*, while the $\mathbf{C_{CTS}}$ comes from the component *c*. The $\mathbf{C_{BN}}$ contains 0.4 hours speech data spoken by 4 speakers (1 female and 3 male speakers), while the $\mathbf{C_{CTS}}$ contains 1.8 hours speech data spoken by 25 speakers (12 female and 13 male speakers).

3.1.2 The Jasmin-CGN corpus

The CGN corpus is restricted to only native Dutch adult speakers, so only using the CGN corpus is not possible to quantify the bias against non-native accents in the ASR systems. Hence in addition to the CGN corpus, I use the Jasmin-CGN corpus which contains non-native accented speech data as well. The Jasmin-CGN corpus [60] is an extension of the CGN corpus and is recorded in the Netherlands and Flanders. I only use the data recorded in the Netherlands, so when mentioning Jasmin-CGN corpus later, I refer to the speech data recorded in the Netherlands.

The jasmin-CGN corpus consists of read and human-machine interaction (HMI) types speech spoken by native speakers with 3 age groups (children, teenagers and older adults) and non-native speakers with 2 age groups (teenagers[3], adults). Furthermore, the non-

[1] `https://github.com/laurensw75/kaldi_egs_CGN`

[2] `N-best2018TestSetsforCGN`

[3] In [60], it claims that these speakers are children (between 7 and 14), but actually they are teenagers (between 11 and 18).

native speakers come from 37 different countries such as Afghanistan, Andorra, Egypt and Spain, etc. Particularly, the general information and the corresponding duration of the raw speech data of the 5 speakers groups in jasmin-CGN corpus are 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.

Normally, each speaker from the 5 groups records 2 types of speech data i.e., read and HMI speech data. However, there are also several speakers (for example, speaker ID "N000337" and speaker ID "N000216" in the NNA group) who record only one type of speech data. The raw speech data in the jasmin-CGN corpus is processed by segmenting into small pieces and then removing the silence parts, using the open-source code[4] given by [9].

After the pre-processing, I get 40.24 hours speech data in total. Next, the speech data is divided into two parts respectively for training and test. The training data (36.12 hours; spoken by 137 female and 104 male speakers) in jasmin-CGN corpus is used to train ASR models denoted by $\mathbf{J_{train}}$. Furthermore, the non-native accented speech data in $\mathbf{J_{train}}$ is also used to train a VC model. The $\mathbf{J_{train}}$ consists of 14.1 hours non-native accented speech (10.42 hours read data and 3.69 hours HMI data) and 22.02 hours native speech (16.31 hours read data and 5.70 hours HMI data).

For each group of speakers (DC, DT, DOA, NNT and NNA), I select 6 speakers (3 female and 3 male speakers) who record both read and HMI speech data for making test sets, resulting in 10 small test set i.e. read or HMI speech data spoken by DC denoted by $\mathbf{R/H_{DC}}$, by DT denoted by $\mathbf{R/H_{DT}}$, by DOA denoted by $\mathbf{R/H_{DOA}}$, by NNT denoted by $\mathbf{R/H_{NNT}}$ and by NNA denoted by $\mathbf{R/H_{NNA}}$, and the detailed speaker information of all the small test sets are listed in tables 3.1, 3.2, 3.3, 3.4 and 3.5 respectively. For the native speakers in the test sets $\mathbf{R/H_{DC}}$, $\mathbf{R/H_{DT}}$ and $\mathbf{R/H_{DOA}}$, I simply list the speaker ID, gender and age information, while for the non-native speakers in the test sets $\mathbf{R/H_{NNT}}$, $\mathbf{R/H_{NNA}}$, I give more information about their birth countries and mother languages.

The 10 small test sets ($\mathbf{R/H_{DC}}$, $\mathbf{R/H_{DT}}$, $\mathbf{R/H_{DOA}}$, $\mathbf{R/H_{NNT}}$ and $\mathbf{R/H_{NNA}}$) are used to measure the speech recognition performance i.e., WERs of the trained ASR models. However, using these 10 small test sets separately are not suitable for quantifying the bias against non-native accents, because except that DT (speakers age 12-18) and NNT (speakers age 11-18) groups are quite similar with the age range, the age ranges in DC and DOA groups have no overlap with the NNA group. Hence I merge the DC, DT and DOA groups to get 2 bigger test sets respectively for native accented read and HMI speech data. I also merge the NNT and NNA groups to get 2 bigger test sets respectively for non-native accented read and HMI speech data. The detail information and notations of the 4 bigger test sets are listed below, which are used to quantify the bias against non-native accents on average.

[4] `https://github.com/syfengcuhk/jasmin`

Table 3.1: Speaker information of test sets $\mathbf{R_{DC}}$ and $\mathbf{H_{DC}}$: Native Dutch children speakers.

Speaker ID	Gender	Age
N000025	Female	8
N000027	Male	9
N000029	Male	10
N000054	Female	11
N000045	Male	12
N000213	Female	7

Table 3.2: Speaker information of test sets $\mathbf{R_{DT}}$ and $\mathbf{H_{DT}}$: Native Dutch teenager speakers.

Speaker ID	Gender	Age
N000240	Female	12
N000251	Male	13
N000254	Female	14
N000267	Male	15
N000271	Female	16
N000276	Male	12

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

3.1.3 The VCTK corpus

The VCTK corpus[61] is an English multi-speaker corpus, consisting of speech from 110 English speakers (47 female and 62 male speakers and 1 speaker with unknown gender) with various accents. The VCTK corpus has around 44 hours speech data. The speech data in the VCTK corpus are quite clear and it is widely used for VC techniques [62] [24] and text-to-speech techniques [63][64] researches. There are about 400 utterances (around 5 seconds for each utterance) selected from a newspaper spoken by each speaker in the VCTK corpus.

Since I only have very limited non-native accented speech data (14.1 hours) in $\mathbf{J_{train}}$. In order to train a good VC model, in addition to the non-native accented speech data in $\mathbf{J_{train}}$, I also use the VCTK corpus and ignore the speaker with unknown gender. Furthermore, I use both the non-native accented speech data $\mathbf{J_{train}}$ and the VCTK corpus to generate more non-native accented speech data through the trained VC model.

Table 3.3: Speaker information of test sets $\mathbf{R}_{\mathrm{DOA}}$ and $\mathbf{H}_{\mathrm{DOA}}$: Native Dutch adult speakers.

Speaker ID	**Gender**	**Age**
N100018	Female	69
N100056	Male	88
N100035	Female	93
N100054	Male	83
N100078	Female	78
N100082	Male	95

Table 3.4: Speaker information of test sets $\mathbf{R}_{\mathrm{NNT}}$ and $\mathbf{H}_{\mathrm{NNT}}$: Non-native teenager speakers.

Speaker ID	**Gender**	**Age**	**Birth country**	**Mother language**
N000221	Female	18	Turkey	Turkish
N000222	Female	16	Iraq	Arabic
N000233	Female	13	Poland	Polish
N000224	Male	18	Nigeria	English
N000225	Male	16	Nigeria	Bini Edo
N000234	Male	13	Afghanistan	Pushto

Table 3.5: Speaker information of sub-test sets $\mathbf{R}_{\mathrm{NNA}}$ and $\mathbf{H}_{\mathrm{NNA}}$: Non-native adult speakers.

Speaker ID	**Gender**	**Age**	**Birth country**	**Mother language**
N000019	Male	23	Czech Republic	Turkish
N000328	Female	27	Russia	Russian
N000259	Female	31	China	Chinese
N000343	Male	37	Meng	English
N000325	Male	49	Afghanistan	Persian
N000338	Female	52	Algeria	Arabic

3.2 Baselines

In order to explore the research questions RQ1 and RQ2, the first step is to train the baselines to quantify the non-nativeness bias in the SOTA ASR models. Using SpeechBrain toolkit, I train 2 E2E ASR models as baselines i.e., RNN-based and transformer-based E2E ASR models based on 2 latest given ASR training codes [5]. I train two ASR models to compare if the bias against non-native accents exists prevalently for different ASR models and to further explore if the methods I proposed for bias mitigation can take effect on different ASR models.

3.2.1 Speech data processing

I use the training data set in the CGN corpus $\mathbf{C_{train}}$ to build the RNN-based and transformer-based baselines. As described in the Section 3.1.1, $\mathbf{C_{train}}$ includes spontaneous telephone dialogues. The sample rate of spontaneous telephone dialogues speech data is 8000Hz, which is different from other speech data with 16000Hz sample rate. In order to train the baselines with all the speech data in $\mathbf{C_{train}}$, I upsample the 8000Hz telephone type speech data to 16000Hz. Meanwhile, the $\mathbf{C_{CTS}}$ test set come from component *c* in the CGN corpus, so it is also 8000Hz. I upsample the speech data in $\mathbf{C_{CTS}}$ to 16000Hz as well.

For both the RNN-based and the transformer-based ASR models, I follow the default feature settings in the SpeechBrain toolkit and mel-sectrograms are used as input feature vectors. For the RNN-based ASR model, the number of Mel bins is 40, while for the transformer-based one, the number of Mel bins is 80.

3.2.2 RNN-based E2E ASR model implementation

The RNN-based E2E ASR model is built upon the open-source code[6].

Experimental set-ups

I train the ASR models with $\mathbf{C_{train}}$ for 11 epochs and test with CGN test sets $\mathbf{C_{BN}}$ and $\mathbf{C_{CTS}}$ to see the WER performances. I quantify the bias against non-native accents of the read and HMI types speech separately with the Jasmin-CGN test sets, as there exists big acoustical and linguistic difference between read and conversational types speech [65][66][31]. The bias is quantified by using test sets $\mathbf{R_N}$ and $\mathbf{R_{NN}}$ for read speech according to Equation 3.1 and using test sets $\mathbf{H_N}$ and $\mathbf{H_{NN}}$ for HMI speech according to Equation 3.2.

$$\mathbf{B_R} = WER(\mathbf{R_{NN}}) - WER(\mathbf{R_N}) \tag{3.1}$$

$$\mathbf{H_R} = WER(\mathbf{H_{NN}}) - WER(\mathbf{H_N}) \tag{3.2}$$

Where $\mathbf{B_R}$ is the bias against non-native accents for read speech data, and $\mathbf{B_H}$ is for HMI speech data.

As mentioned in Section 3.1.2, the $\mathbf{R_N}$, $\mathbf{R_{NN}}$, $\mathbf{H_N}$ and $\mathbf{H_{NN}}$ are composed of several small test sets. Hence I also test with the 10 small test sets $\mathbf{R/H_{DC}}$, $\mathbf{R/H_{DT}}$, $\mathbf{R/H_{DOA}}$, $\mathbf{R/H_{NNT}}$ and $\mathbf{R/H_{NNA}}$ to see the detailed WER performances (these detailed WER results are included in Appendix 6).

[5] The codes are selected as it claims that the models trained using such codes can obtain the SOTA performance on LibriSpeech [27] English data set.

[6] RNN-basedASR

Configurations of the ASR model

The configurations of the RNN-based ASR model are given below:

- **Tokenizer:** I use 5000 byte pair encoding (BPE) tokens trained on all the transcriptions of the speech data in $\mathbf{C_{train}}$.
- **Speech recognizer:** The encoder is CRDNN (combinations of CNNs, RNNs and DNNs) consisting of 2 CNN blocks, 4-layer RNNs and 2 DNN blocks. The decoder is one-layer GRU. It uses the joint CTC/attention mechanism to decode.

3.2.3 Transformer-based E2E ASR model implementation

The transformer-based E2E ASR model is built upon the open-source code[7].

Experimental set-ups

The experimental set-ups is the same as that in Section 3.2.2.

Configurations of the ASR model

The configurations of the transformer-base ASR model are given below:

- **Tokenizer:** I use 5000 byte pair encoding (BPE) tokens trained on all the transcriptions of the speech data in $\mathbf{C_{train}}$.
- **Speech recognizer:** The encoder consists of 2 CNN blocks (each CNN block contains one CNN layer) and a 12-layer transformer. The decoder consists of a 6-layer transformer. It uses joint CTC/attention mechanism to decode.

3.3 Data augmentation

I train the baselines with the data in CGN corpus, and quantify the bias by testing with the data from another database i.e., jasmin-CGN corpus. I hypothesis that the mismatch between the training data (only native accented speech data) and the test data (both native and non-native accented speech data) can be the reason where the bias come from. Hence, I merge the speech data in $\mathbf{J_{train}}$ and $\mathbf{C_{train}}$ together to retrain the baseline ASR models with the same experimental set-up and configurations as baselines described in Section 3.2. Now, in the training data, I have 402.14 hours native accented speech data in total but only 14.1 hours non-native accented speech data (as mentioned in Section 3.1.1 and 3.1.2, all the 380.12 hours speech data in $\mathbf{C_{train}}$ is native accented while there are 22.02 hours native accented speech data and 14.1 hours non-native accented speech data in $\mathbf{J_{train}}$). Even though I add $\mathbf{J_{train}}$ to the training data, the training data set is still very imbalanced about native and non-native accented speech data i.e., the lack of non-native accented speech data. Hence data augmentation is only applied for the non-native accented speech data in $\mathbf{J_{train}}$. The detailed non-native accented speech data augmentation implementations are illustrated in Section 3.3.1 for the VC-based method and in Section 3.3.2 for the speed perturbation method. After generating more non-native accented speech data, I conduct experiments using the generated data to answer the RQ1 and the corresponding experimental set-ups are described in Section 3.3.3..

[7] `Transformer-basedASR`

3.3.1 Cross-lingual VC implementation

For the VC experiments, I use a SOTA non-parallel VC model: AGAIN-VC [24][8].

Speech data processing

For VC experiments, the speech data in the VCTK corpus and $\mathbf{J_{train}}$ is processed following [24]. All the speech data is downsampled or upsampled to 22050Hz. After which, silence at the start and end of each audio clip are removed. Next, the mel-spectrogram features of the audio clips are extracted with 80 mel bins.

Experimental set-ups

I train two VC models respectively for non-native accented read and HMI types speech data augmentation. For the HMI speech data, I use the VCTK corpus and the non-native accented HMI speech data in the $\mathbf{J_{train}}$ as the training data. For the read speech data, I use the VCTK corpus and the non-native accented read speech data in the $\mathbf{J_{train}}$ as the training data. For both VC models, I train 100000 steps with 32 batch size.

I separate the generation of non-native accented read and HMI types speech data, because those two types speech data have big difference even they are spoken by the same speaker. Furthermore, as mentioned in Section 3.1.2, not all the speakers in $\mathbf{J_{train}}$ have two types speech data. Only training one VC model for both non-native accented read and HMI types speech data may also work and it can be the future work.

In this thesis, the VC model trained with HMI data is used for non-native accented HMI speech data augmentation, while the one trained with read data is for read speech. Specifically, I convert the voices of the non-native Dutch speakers from $\mathbf{J_{train}}$ to those of the English speakers in the VCTK corpus. The reason is that it is needed to ensure that the generated VC speech contains non-native accented speech characteristics. Converting Dutch native speech as the source while using the Jasmin non-native accented speech or the VCTK English speakers as the target will not ensure this. As the voice conversion is not the same as accent conversion. To be more specific when I use the native accented speech as the source data, the voice of the generated speech is different from the source speech data while the accent of the generated speech may still have similarities with that of the source speech data. For the way I do the VC, even though the accent of the generated speech has similarities with that of the source speech data, now the accent of the source speech data spoken by English speakers is still non-native for Dutch. In addition, I try voice conversions among the non-native speakers, but the generated speech sounds not clear at all.

The detailed inference steps are listed below:

1. **Load the VC model:** I trained two VC models for read and HMI types speech respectively. When I aim for generating more non-native accented read speech data, I use the VC model trained with that type of data (the same process for non-native accented HMI speech data).

2. **Pick source-target pairs:** given the source speakers and the target speakers, it is needed to decide that I convert the speech data from which source speaker to the

[8] `https://github.com/KimythAnly/AGAIN-VC`

speech data from which target speaker. It is possible to randomly choose source-target speaker pair. I assume that the more similar the source speaker is to target speaker, the better the quality of the generated speech data. Based on the assumption, I use the cosine speaker similarity method [67] to pick source-target pairs. Using the pre-trained ConvGRU speaker encoder [9] to get the embeddngs (vectors) for both target and source speakers, and then calculate the cosine similarity of each possible source-target speaker pair. I pick the source-target pair of which cosine similarity is greater or equal to 0.2 to generate more non-native accented speech data for both read and HMI types speech data. Note when I pick the source-target pair for generating non-native accented read speech data, I use the read type speech to calculate the speaker similarity (the same process for non-native accented HMI speech data).

3. **Convert the mel-spectrogram to speech data:** the output of the AGAIN-VC model is the mel-spectrogram of the converted speech data. As in [24], the output mel-spectrogram of the VC model is re-synthesised using a pre-trained Mel-GAN vocoder and I downsample the generated speech data to 16000Hz for the subsequent non-nativeness bias mitigation experiments.

As a result, 19.13 hours of non-native read data denoted by $\mathbf{vc_{rd}}$ and 4.25 hours of non-native HMI data denoted by $\mathbf{vc_{hmi}}$ are generated. The amount of speech data in $\mathbf{vc_{hmi}}$ and $\mathbf{vc_{rd}}$ are quite different. There are two main reasons: for non-native accented speech data in $\mathbf{J_{train}}$, the amount of read and HMI types speech are different; Moreover, the source-target speaker pairs for read and HMI types speech are different. I use $\mathbf{vc_{all}}$ to denote the combination of the speech data in $\mathbf{vc_{hmi}}$ and $\mathbf{vc_{rd}}$.

3.3.2 Speed perturbation implementation

I used the standard speech perturbation[20] data augmentation method: the speed command of *sox* is used to do two-fold speed perturbation data augmentation (with 0.9 and 1.1 perturbation factors) for both the non-native accented read speech data and HMI data in $\mathbf{J_{train}}$, respectively denoted by $\mathbf{sp_{rd}}$ and $\mathbf{sp_{hmi}}$. $\mathbf{sp_{all}}$ indicates the combination of $\mathbf{sp_{rd}}$ and $\mathbf{sp_{hmi}}$.

3.3.3 Experimental set-ups for augmented speech applications

In order to answer the RQ1, I use the augmented non-native accented speech data to conduct experiments: adding the augmented speech data to the traing data and then retraining the ASR models (for 11 epochs) with the same architectures used in baselines. The experimental set-ups for test are the same as that of baselines described in Section 3.2.2. The experimental set-ups of training data are listed below:

- **Cross-lingual VC for non-native accented read/HMI data:** use $\mathbf{vc_{rd}}$ and/or $\mathbf{vc_{hmi}}$, $\mathbf{C_{train}}$, and $\mathbf{J_{train}}$ as the training data to train ASR models,
- **Speed perturbation for non-native accented read/HMI data:** use $\mathbf{sp_{rd}}$ and/or $\mathbf{sp_{hmi}}$, $\mathbf{C_{train}}$, and $\mathbf{J_{train}}$ as the training data to train ASR models,

[9] https://github.com/RF5/simple-speaker-embedding

- **Combinations of the VC augmented and the speed perturbed speech data:** use $\mathbf{vc_{all}}$, $\mathbf{sp_{all}}$, $\mathbf{C_{train}}$, and $\mathbf{J_{train}}$ to train the better ASR model used in 2 baselines to explore if this setting can lead to better de-biasing results.

3.4 ASR Training strategies

In order to answer the RQ2, I conduct experiments based on the baselines but with different training strategies i.e., fine-tuning and DAT. The baselines are seemed as the standard training method. In addition to the speech data in $\mathbf{J_{train}}$, I also use the generated non-native accented speech data ($\mathbf{vc_{rd}}$, $\mathbf{vc_{hmi}}$, $\mathbf{sp_{rd}}$, $\mathbf{sp_{hmi}}$) to conduct experiments with fine-tuning and DAT, which further explores the RQ1 with the different training strategies. In this section, the speech data processing is the same as that in 3.2.1 and the test settings are the same as that in Section 3.2.2.

3.4.1 Fine-tuning implementation

The amount of speech data between native and non-native accented speech data can be more balanced when only using $\mathbf{J_{train}}$ to build an ASR system. Hence it is possible to obtain an ASR system with low bias against non-native accents (the exploration for this hypothesis are included in Appendix 6). However, the total duration of the speech data in $\mathbf{J_{train}}$ may be too little to build a satisfying ASR system. With the help of fine-tuning, I can take the speech knowledge in the big standard native accented speech training set i.e., $\mathbf{C_{train}}$ and fine-tune on a relatively balanced (in terms of the amount of native accented speech data and the amount of the non-native accented speech data) training sets. Thus I conduct fine-tuning experiments to find answers for RQ2.

I have 2 baselines i.e., RNN-based and transformer-based. The RNN-based baseline (trained on $\mathbf{C_{train}}$) is used as the pre-trained ASR model for the RNN-based ASR fine-tuning experiments. Meanwhile, the transformer-based baseline (trained on $\mathbf{C_{train}}$) is then used as the pre-trained ASR model for the transformer-based ASR fine-tuning experiments. For both ASR architectures, I continue to train the corresponding pre-trained model for 5 epochs on different combinations of the speech data in $\mathbf{J_{train}}$ and the generated non-native accented speech data.

For both RNN-based and transformer-based ASR model, I conduct fine-tuning experiments on the training set $\mathbf{J_{train}}$ first and then the fine-tuned ASR models are tested as the test experimental set-ups in Section 3.2.2. Based on the WER and bias results, I choose the better ASR model to conduct the fine-tuning experiment with $\mathbf{J_{train}}$ and all the available generated non-native accented speech data i.e., $\mathbf{vc_{all}}$ and $\mathbf{sp_{all}}$ to pursue the better bias mitigation performance as well as answer the RQ1 about "with the fine-tuning training method, if the data augmentation methods used in this thesis are still helpful for non-nativeness bias mitigation ".

3.4.2 DAT implementation

To investigate the effect of DAT on bias mitigation, a domain classifier is added to the ASR models used in baselines. The ASR models i.e., RNN-based and transformer-based combined with DAT are respectively shown in Figures 3.2 and 3.3. From Figures 3.2 and 3.3, when the DAT combines with the two ASR models, there is no difference other than the structures of the ASR models themselves.

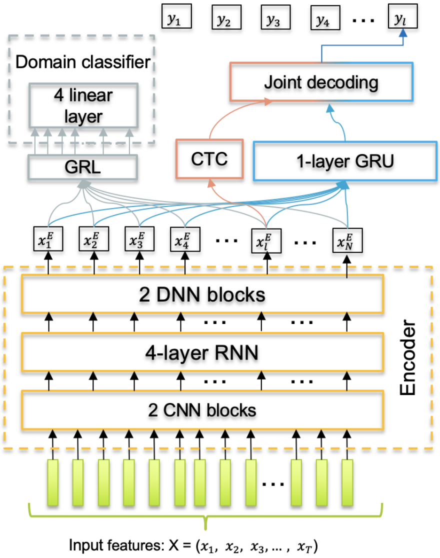

Figure 3.2: The RNN-based ASR model combined with DAT.

The domain classifier (see top-left of the Figures 3.3 and 3.3) is a binary classifier composed of 4 linear layers which share the same features with the decoder. It is used to classify whether the input data is spoken by a native speaker or a non-native speaker. With the help of a gradient reversal layer (GRL), the features extracted by the encoder is able to be adjusted, making the features accent-invariant.

In addition, for all the experiments in this section, I use supervised-DAT and combine it with the baselines (I make the corresponding code open-sourced [10] [11].), which means that except for training the ASR model with native speech data. I also use the non-native accented speech data to train the ASR model because the transcriptions of the non-native accented speech data are available, and the experimental results in [29] suggest speech recognition performance benefits from the supervised-DAT.

For both ASR models combined with DAT, I train them for 11 epochs. When ASR models are combined with DAT, the total loss is calculated as equation 3.3, consisting of the loss of the ASR model denoted by $\mathcal{L}_{\text{ASR}}$ and the loss of the domain classifier denoted by $\mathcal{L}_{\text{domain}}$.

$$\mathcal{L}_{\text{DAT}} = \mathcal{L}_{\text{ASR}} + \lambda \mathcal{L}_{\text{domain}} \tag{3.3}$$

where $\lambda \in \mathbb{R}$ is a hyper-parameter, and it controls the degree of influence of the domain classifier on the whole model.

To begin with, I set λ to 0.01, the same value as [18] (also work for non-native accented speech data but in English). I use $\mathbf{C}_{\text{train}}$ and $\mathbf{J}_{\text{train}}$ as the training data for both ASR models,

[10] https://github.com/Yuanyuan-888/supervided-DAT-rnn-recipe
[11] https://github.com/Yuanyuan-888/supervided-DAT-transformer-recipe

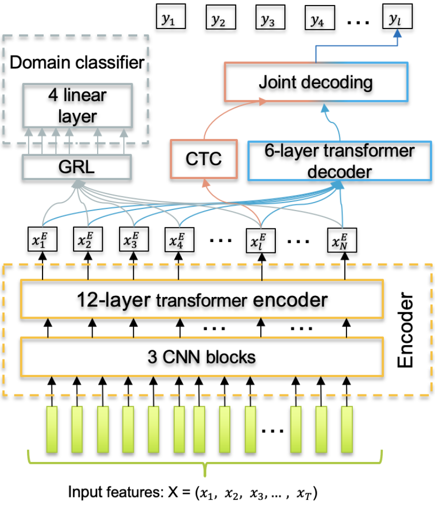

Figure 3.3: The transformer-based ASR model combined with DAT.

so as to compare their performances when combined with DAT. After then, I choose the better ASR model combined with DAT to conduct more experiments with $\mathbf{C}_{\text{train}}$, $\mathbf{J}_{\text{train}}$ and all the available generated non-native accented speech data i.e., $\mathbf{vc}_{\text{all}}$ and $\mathbf{sp}_{\text{all}}$ to pursue better bias mitigation results and find answers about RQ1 "with the DAT training method, if the data augmentation methods used in this thesis are still helpful for non-nativeness bias mitigation".

According to the previous researches in [29] and [18], for different databases, the optimal value of λ are usually different. Consequently, I choose the training data setting with best bias mitigation performance and relatively low WERs and do more experiments to compare and find the better value of λ for bias mitigation in my settings. I conduct 10 more experiments with 10 different λ values which are respectively 0.01, 0.02, 0.03, 0.04, 0.05, 0, 06, 0.07, 0, 08, 0.09, 0.1 (the corresponding experimental results are shown in Appendix 6).

4

Results

In this chapter, first, the experimental results i.e. the bias quantification for read and HMI types speech data of the 2 baselines are described in Section 4.1. Moreover, in order to exploring the RQ1, I carry out the data augmentation experiments and describe the experimental results in Section 4.2. Besides, for exploring the RQ2, I also carry out the effective training strategies experiments and describe the experimental results in Section 4.3.

Table 4.1: Experimental results: WERs on $C_{CTS/BN}$ and $R/H_{N/NN}$ and the non-nativeness bias quantification (for both read and HMI types speech). All the numbers with bold notifications means the best value in each column. **BL**: baseline; **TF-based**: transformer-based; **F-T**: fine-tuning.

Details		**CGN (% WER)**		**Jasmin (% WER and Bias)**					
Model	**Train data**	C_{BN}	C_{CTS}	R_N	R_{NN}	H_N	H_{NN}	B_R	B_H
BL: RNN-based	C_{train}	16.84	43.55	34.18	63.25	39.84	68.49	29.07	28.65
BL: TF-based	C_{train}	9.64	37.99	24.9	53.73	30.77	60.26	28.83	29.49
RNN-based	C_{train},J_{train}	23.19	45.41	9.97	28.39	30.18	42.42	18.42	12.24
TF-based	C_{train},J_{train}	9.75	37.01	5.37	21.16	20.12	36.50	15.79	16.38
RNN-based	C_{train},J_{train}, vc_{all}	21.68	44.64	11.71	29.53	30.1	44.26	17.82	14.16
TF-based	C_{train},J_{train}, vc_{all}	9.46	37.02	4.88	18.83	20.80	34.32	13.95	13.52
TF-based	C_{train},J_{train}, vc_{rd}	10.84	38.99	5.31	19.49	21.09	37.90	14.18	16.81
TF-based	C_{train},J_{train}, vc_{hmi}	9.50	37.45	5.55	20.48	20.53	35.43	14.93	14.90
RNN-based	C_{train},J_{train}, sp_{all}	19.86	46.08	10.87	28.21	30.91	43.05	17.34	12.14
TF-based	C_{train},J_{train}, sp_{all}	9.59	36.72	4.98	18.75	20.22	33.16	13.77	12.94
TF-based	C_{train},J_{train}, sp_{rd}	10.08	37.54	4.90	18.78	20.77	35.89	13.88	15.12
TF-based	C_{train},J_{train}, sp_{hmi}	9.61	37.25	5.28	20.70	20.45	34.20	15.42	13.75
TF-based	C_{train},J_{train}, vc_{all}, sp_{all}	**9.27**	**36.53**	4.79	**18.38**	**19.62**	**32.24**	13.59	12.62
F-T: RNN-based	J_{train}	43.45	55.57	8.09	24.96	29.91	41.89	16.87	11.98
F-T: TF-based	J_{train}	30.93	48.51	5.00	20.42	21.27	35.26	15.42	13.99
F-T: TF-based	J_{train}, vc_{all}, sp_{all}	43.76	55.74	**4.78**	19.25	23.90	34.03	14.47	**10.13**
DAT: RNN-based	C_{train},J_{train}	17.57	54.54	17.23	36.72	36.05	57.19	19.49	21.14
DAT: TF-based	C_{train},J_{train}	11.00	40.89	6.12	21.81	22.25	38.77	16.52	15.69
DAT: TF-based	C_{train},J_{train}, vc_{all}, sp_{all}	10.23	38.40	4.86	18.40	20.24	32.87	**13.54**	12.63

4.1 Baselines

Before exploring the main RQ: "the methods for mitigating the bias against non-native accents", it is needed to first know the original non-nativeness bias for read and HMI types speech in the 2 baselines i.e. RNN-based and transformer-based ASR models. The rows with "**BL: RNN-based**" and "**BL: Transformer-based**" in the Table 4.1 show the overall results of the baselines in terms of the speech recognition performance and the bias $\mathbf{B_R}$ and $\mathbf{B_H}$.

First, for the row with "**BL: RNN-based**" in the Table 4.1, in terms of the speech recognition performance on CGN test sets, the WER of $\mathbf{C_{CTS}}$ is worse than that of $\mathbf{C_{BN}}$; and in terms of the speech recognition performance on Jasmin-CGN test sets, the WER of $\mathbf{H_N}$ is worse than that of $\mathbf{R_N}$, and the WER of $\mathbf{H_{NN}}$ is worse than that of $\mathbf{R_{NN}}$. For both in-domain test sets and the out-domain test sets for the RNN-based baseline, the read type speech data has better recognition performance than the conversational type speech data, which is not surprising. For the bias against non-native accents, overall the RNN-based baseline show big bias against non-native accents for both the read and HMI types speech. Moreover, the bias for the read speech data $\mathbf{B_R}$ is quite similar with that for the HMI speech data $\mathbf{B_H}$ with only 0.42% bias difference.

Second, for the row with "**BL: Transformer-based**" in the Table 4.1, for both the CGN and jasmin-CGN test sets, the test sets $\mathbf{C_{BN}}$, $\mathbf{R_N}$ and $\mathbf{R_{NN}}$ with the read type speech data all have better recognition performance than the test sets $\mathbf{C_{CTS}}$, $\mathbf{H_N}$ and $\mathbf{H_{NN}}$ with the HMI type speech data respectively, the same as the results of "**BL: RNN-based**". In terms of the bias against non-native accents, different from the results of "**BL: RNN-based**", for "**BL:**

Transformer-based", the bias for HMI speech data $\mathbf{B_H}$ is a little bit smaller than that for read speech data $\mathbf{B_R}$, with only 0.66% bias difference.

In short, regarding to the bias against non-native accents, for both RNN-based and transformer-based baselines, they do not have obvious bias value gap, and the bias for both read and HMI types speech data mainly comes from the bad non-native accented speech recognition performance. For WERs of all the test sets listed in the Table 4.1, the transformer-based baseline outperforms the RNN-based baseline a lot.

4.2 Data augmentation

4.2.1 Merge training sets in the CGN and jasmin-CGN corpora

Before adding the generated accented speech data to retrain the ASR models, first, I merge $\mathbf{J_{train}}$ and $\mathbf{C_{train}}$ together to retrain the 2 ASR models. Furthermore, the results for merging $\mathbf{J_{train}}$ and $\mathbf{C_{train}}$ experiments are shown in the Table 4.1 (with $\mathbf{C_{train}}$, $\mathbf{J_{train}}$ in the "**train data**" column).

For the RNN-based ASR model, using $\mathbf{C_{train}}$ and $\mathbf{J_{train}}$ as the training data, compared the results with only using $\mathbf{C_{train}}$ as the training data, the main differences are:

- For the test sets in the CGN i.e., $\mathbf{C_{BN}}$ and $\mathbf{C_{CTS}}$, their WERs become higher,
- For the test sets in the Jasmin-CGN i.e., $\mathbf{R_N}$, $\mathbf{R_{NN}}$, $\mathbf{H_N}$, and $\mathbf{H_{NN}}$, their WERs become lower a lot,
- As for the bias $\mathbf{B_R}$ and $\mathbf{B_H}$, they reduce a lot. For the value of $\mathbf{B_R}$, it reduces by 10.65%, while for the value of $\mathbf{B_H}$, it reduces by 16.41%, leading to a relatively big gap (6.18%) between the bias for read speech and that for HMI speech.

For the transformer-based ASR model, using $\mathbf{C_{train}}$ and $\mathbf{J_{train}}$ as the training data, compared the results with only using $\mathbf{C_{train}}$ as the training data, the main differences are:

- The performance on test sets in Jasmin-CGN are improved a lot,
- The bias $\mathbf{B_R}$ and $\mathbf{B_H}$ reduce a lot. For the value of $\mathbf{B_R}$, it reduces by 13.03%, while for the value of $\mathbf{B_N}$, it reduces by 13.11%.

Overall, comparing the "**RNN-based**" and the "**transformer-based**" experiments with the "$\mathbf{C_{train}}$, $\mathbf{J_{train}}$" training data setting, for all WERs of the test sets listed in the Table 4.1, the transformer-based ASR model performs better than the RNN-based one. For the non-nativeness bias, the value of $\mathbf{B_R}$ in the RNN-based ASR system is 2.63% bigger than that in the transformer-based one, while the value of $\mathbf{B_H}$ in the RNN-based experiment is 4.14% smaller than that in the transformer-based one.

4.2.2 Cross-lingual VC-based data augmentation

$\mathbf{vc_{all}}$ experiments

The experimental results of adding $\mathbf{vc_{all}}$ to retrain the ASR models are shown in Table 4.1 (with $\mathbf{C_{train}}$,$\mathbf{J_{train}}$,$\mathbf{vc_{all}}$ in the "**Train data**" column).

For the RNN-based ASR model, using $\mathbf{C_{train}}$, $\mathbf{J_{train}}$ and $\mathbf{vc_{all}}$ as the training data, compared the results with using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ as the training data, the main differences are:

- For the test sets in the CGN i.e. $\mathbf{C_{BN}}$ and $\mathbf{C_{CTS}}$, the WERs are slightly improved i.e. the WERs become lower, however, if compared with the CGN test results of the RNN-based baseline: "**BL: RNN-based**", the WERs are still degraded i.e. the WERs are still higher,
- For the test sets in the Jasmin-CGN, except that the WER of $\mathbf{H_N}$ nearly does not change, while the values of WERs on the other 3 test sets $\mathbf{R_N}$, $\mathbf{R_{NN}}$ $\mathbf{H_{NN}}$ are respectively 1.74%, 1.14% and 1.84% higher,
- As for the value of $\mathbf{B_R}$, it reduces, however, at the expense of recognition performance on the read native accented speech data $\mathbf{R_N}$; as for the value of $\mathbf{B_H}$, it becomes bigger because of the degration of the HMI non-native accented speech recognition.

For the transformer-based ASR model, using $\mathbf{C_{train}}$, $\mathbf{J_{train}}$ and $\mathbf{vc_{all}}$ as the training data, compared the results with using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ as the training data, the main differences are:

- For the test sets in Jasmin-CGN corpus, except that the value of WER on the H_N becomes bigger slightly, however, it is still far more better than the WER in the transformer-based baseline. All the performance of the other 3 test sets continue to improve.
- The bias $\mathbf{B_R}$ and $\mathbf{B_H}$ continue to reduce. For the value of $\mathbf{B_R}$, it reduces by 1.84%, while for the value of $\mathbf{B_N}$, it reduces by 2.86%.

Overall, just using the standard training method, the cross-lingual VC-based non-native accented speech data augmentation does not work for non-native accents bias mitigation in the RNN-based ASR system, but works in the transformer-based ASR system well.

$\mathbf{vc_{rd}}$ AND $\mathbf{vc_{hmi}}$ EXPERIMENTS

Since cross-lingual VC-based non-native accented speech data augmentation works for bias mitigation in the transformer-based ASR system, next, I conduct cross-lingual VC-based data augmentation experiments for read and HMI speech data spoken by non-native speakers separately with the transformer-based ASR system for answering the RQ1. The experimental results are listed in the Table 4.1 (with $\mathbf{C_{train}}$,$\mathbf{J_{train}}$,$\mathbf{vc_{rd}}$ or $\mathbf{C_{train}}$,$\mathbf{J_{train}}$,$\mathbf{vc_{hmi}}$ in the "**Train data**" column).

For the $\mathbf{C_{train}}$,$\mathbf{J_{train}}$,$\mathbf{vc_{rd}}$ experiment, compared with the experimental results of using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ as the training data with the transfromer-based ASR model, the main differences are:

- For the tests sets in the CGN, both the WERs on the $\mathbf{C_{BN}}$ and $\mathbf{C_{CTS}}$ are respectively 1.09% and 1.98% higher,
- For the test sets in the Jasmin-CGN, the WERs of read speech test sets $\mathbf{R_N}$ and $\mathbf{R_{NN}}$ become lower, while the WERs of HMI speech test sets $\mathbf{H_N}$ and $\mathbf{H_{NN}}$ become higher, which means that adding the $\mathbf{vc_{rd}}$ as the training data brings benefits for the read type speech data,
- For the bias $\mathbf{B_R}$, it is 1.61% smaller, while for the bias $\mathbf{B_H}$, it is worse i.e. it is bigger.

For the $\mathbf{C_{train}}$,$\mathbf{J_{train}}$,$\mathbf{vc_{hmi}}$ experiment, compared with the experimental results of using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ as the training data with the transfromer-based ASR model, the main differences are:

- For the test sets in the Jasmin-CGN, the values of WERs of non-native accented speech $\mathbf{R_{NN}}$ and $\mathbf{H_{NN}}$ are respectively 0.68% and 1.07% lower,

- For the values of bias $\mathbf{B_R}$ and $\mathbf{B_H}$, they are respectively 0.86% and 1.61% smaller.

Overall, VC-based non-native accented read speech data augmentation brings the debiasing result for the read speech data, while VC-based non-native accented HMI speech data augmentation brings debiasing results for both the read and HMI types speech data (but leads to better performance for HMI accented speech bias mitigation). Furthermore, using the combination of $\mathbf{vc_{rd}}$ and $\mathbf{vc_{hmi}}$ data i.e. $\mathbf{vc_{all}}$ is better than using $\mathbf{vc_{rd}}$ and $\mathbf{vc_{hmi}}$ separately, nearly for all evaluation metrics listed in Table 4.1.

4.2.3 Speed perturbation data augmentation

$\mathbf{sp_{all}}$ experiments

The experimental results of adding $\mathbf{sp_{all}}$ to retrain the ASR systems are shown in Table 4.1 (with $\mathbf{C_{train}}$,$\mathbf{J_{train}}$,$\mathbf{sp_{all}}$ in the "**Train data**" column).

For the RNN-based ASR model, using $\mathbf{C_{train}}$, $\mathbf{J_{train}}$ and $\mathbf{sp_{all}}$ as the training data, compared the results with using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ as the training data, the main differences are:

- For the test sets in the CGN i.e. $\mathbf{C_{BN}}$ and $\mathbf{C_{CTS}}$, the WERs are improved i.e. the WERs become lower, However, if compared with the CGN test results of the RNN-based baseline: "**BL: RNN-based**", the WERs are still degraded i.e. the WERs are still higher,

- For the test sets in the Jasmin-CGN, except that the WER of $\mathbf{R_{NN}}$ nearly does not change, while the values of WERs on the other 3 test sets $\mathbf{R_N}$, $\mathbf{H_N}$ $\mathbf{H_{NN}}$ are respectively 0.90%, 0.73% and 0.63% higher,

- As for the value of $\mathbf{B_R}$, it reduces, however, at the expense of recognition performance on the read native accented speech data $\mathbf{R_N}$; as for the value of $\mathbf{B_H}$, it becomes a little bit smaller by 0.1%, meanwhile, showing the degration of both the HMI and read types non-native accented speech recognition.

For the transformer-based ASR model, using $\mathbf{C_{train}}$, $\mathbf{J_{train}}$ and $\mathbf{sp_{all}}$ as the training data, compared the results with using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ as the training data, the main differences are:

- For the test sets in Jasmin-CGN corpus, except that the value of WER on the H_N becomes higher slightly, however, it is still far more better than the WER in the transformer-based baseline. All the performance of the other 3 test sets show improvements.

- The bias $\mathbf{B_R}$ and $\mathbf{B_H}$ continue to reduce. For the value of $\mathbf{B_R}$, it reduces by 2.02%, while for the value of $\mathbf{B_N}$, it reduces by 3.44%.

Overall, just using the standard training method, the speed perturbation non-native accented speech data augmentation leads to small non-native accents bias reduction in the RNN-based ASR system, however, at the expense of speech recognition performance for all Jasmin-CGN test sets listed in Table 4.1. Different from the poor bias mitigation results of speed perturbation data augmentation in the RNN-based ASR system, the speed perturbation data augmentation works well for bias reduction in the transformer-based ASR system.

$\mathbf{sp_{rd}}$ AND $\mathbf{sp_{hmi}}$ EXPERIMENTS

Since speed perturbation-based non-native accented speech data augmentation works for bias mitigation in the transformer-based ASR system better rather than in the RNN-based one, next, I conduct data augmentation experiments for read and HMI speech data spoken by non-native speakers separately with the transformer-based ASR system for answering the RQ1. The experimental results are listed in the Table 4.1 (with $\mathbf{C_{train}}$,$\mathbf{J_{train}}$, $\mathbf{sp_{rd}}$ or $\mathbf{C_{train}}$,$\mathbf{J_{train}}$, $\mathbf{sp_{hmi}}$ in the "**Train data**" column).

For the $\mathbf{C_{train}}$,$\mathbf{J_{train}}$, $\mathbf{sp_{rd}}$ experiment, compared with the experimental results of using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ as the training data with the transformer-based ASR model, the main differences are:

- For the tests sets in the CGN, both the WERs on the $\mathbf{C_{BN}}$ and $\mathbf{C_{CTS}}$ are respectively 0.33% and 0.53% higher,

- For the test sets in the Jasmin-CGN, the values of WERs of read speech test sets $\mathbf{R_N}$ and $\mathbf{R_{NN}}$ become lower, with a reduction by 0.47% and 2.98%, while for the HMI speech test sets, the value of WER of $\mathbf{H_N}$ becomes a bit worse and that of $\mathbf{H_{NN}}$ becomes better,

- For the bias $\mathbf{B_R}$, its value is 1.91% smaller, and for the bias $\mathbf{B_H}$, its value is 1.26% smaller.

For the $\mathbf{C_{train}}$,$\mathbf{J_{train}}$, $\mathbf{sp_{hmi}}$ experiment, compared with the experimental results of using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ as the training data with the transfromer-based ASR model, the main differences are:

- For the test sets in the Jasmin-CGN, except that the WER value of test set $\mathbf{H_{NN}}$ is 2.30% lower, the WER values for all the other 3 test sets do not show obvious changes, leading to the bias reduction of B_H,

- For the values of bias $\mathbf{B_R}$ and $\mathbf{B_H}$, they are respectively 0.37% and 2.63% smaller.

Overall, the same as cross-lingual VC based experiments illustrated in Section 4.2.2, for the transformer-based ASR system, the $\mathbf{sp_{rd}}$ mainly benefits for read non-native accented speech data bias reduction, while $s\mathbf{sp_{hmi}}$ mainly benefits for HMI non-native accented speech data bias reduction. Until now, all the non-native accented speech data augmentation experiments lead to the WER degration of the test set $\mathbf{H_N}$. Moreover, adding the $\mathbf{sp_{rd}}$ and $\mathbf{sp_{hmi}}$ to the training data separately is not as good as using them together i.e. using $\mathbf{sp_{all}}$.

4.2.4 Combine the VC-generated and the speed perturbed speech data

Since both data augmentation methods used in this thesis perform well in the transformer-based ASR system, rather than the RNN-based one, in order to promote the better bias mitigation performance, I conduct the experiment: adding both $\mathbf{vc_{all}}$ and $\mathbf{sp_{all}}$ to the training data to retrain the ASR model. The corresponding experimental results is listed in Table 4.1 (see the row with $\mathbf{C_{train}}$,$\mathbf{J_{train}}$, vc_{all}, sp_{all} in the "**Train data**" column), showing better bias mitigation performance than using $\mathbf{vc_{all}}$ and $\mathbf{sp_{all}}$ separately. Specifically, compared with the experimental results of using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ as the training data with the transfromer-based ASR model, both the speech recognition performance on the test sets from the CGN corpus and from the Jasmin-CGN corpus are improved a lot. Compared with the speech recognition performance on the native accented speech test sets in the Jasmin-CGN corpus i.e. $\mathbf{R_N}$ and $\mathbf{H_N}$, the speech recognition performance on the non-native accented speech test sets $\mathbf{R_{NN}}$ and $\mathbf{H_{NN}}$ improved more, directly leading to big non-nativeness bias reduction for both read and HMI types speech respectively.

4.3 ASR training strategies

4.3.1 Fine-tuning

For all the fine-tuning experiments in this section, as mentioned in Chapter 3.4.1, the pre-trained model is the corresponding baselines i.e. either the RNN-based baseline or the transformer-based baseline.

Fine-tuning with $\mathbf{J_{train}}$

For both the RNN-based and transformer-based ASR models, the experimental results of fine-tuning on $\mathbf{J_{train}}$ are shown in Table 4.1 (with "**F-T**" in the "**Model**" column and $\mathbf{J_{train}}$ in the "**Train data**" column).

For the RNN-based fine-tuning experiment, compared with the experimental results of the standard training method using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ (for RNN-based model), the main differences are:

- For the CGN test sets, the fine-tuning training method leads huge degration in terms of WERs. To be more specific, for test set $\mathbf{C_{BN}}$, the WER increases a lot, from 23.19% to 45.51%. For test set $\mathbf{C_{BN}}$, the WER increases from 45.51% to 55.57%,

- For the Jasmin-CGN test sets, the speech recognition performance is improved. Especially, for the test set $\mathbf{R_{NN}}$, the WER is 3.43% lower.

- For the bias mitigation performance, the fine-tuning training method performs better. For read speech data, the value of $\mathbf{B_R}$ is 1.65% smaller; for HMI speech data, the value of $\mathbf{B_H}$ does not show obvious changes (only 0.26% reduction).

For the transformer-based fine-tuning experiment, compared with the experimental results of the standard training method using $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ (for transformer-based model), the main differences are:

- For the CGN test sets, huge WERs degration happens for both $\mathbf{C_{BN}}$ and $\mathbf{C_{CTS}}$,

- For the Jasmin-CGN test sets, except for the WER imrprovement for test set $\mathbf{H_N}$, the WERs of all the other 3 test sets becomes slightly lower,
- For the bias mitigation performance, the value of $\mathbf{B_H}$ perfectly reduces by 2.39%. while bias mitigation for read speech is at the expense of HMI native accented speech recognition performance.

Generally, with the training data $\mathbf{J_{train}}$, the fine-tuning training method does help with non-nativeness bias mitigation for both RNN-based and transformer-based ASR models, however, at the expense of test sets speech recognition performance in the CGN corpus.

Fine-tuning with $\mathbf{J_{train}}$, $\mathbf{vc_{all}}$, $\mathbf{sp_{all}}$

I conduct the fine-tuning experiment with the transformer-based ASR model, using $\mathbf{J_{train}}$, $\mathbf{vc_{all}}$ and $\mathbf{sp_{all}}$. The corresponding experimental results are listed in Table 4.1 (see the row with both **F-T: TF-based** and $\mathbf{J_{train}}$, $\mathbf{vc_{all}}$, $\mathbf{sp_{all}}$).

Compared with the fine-tuning experimental results with only $\mathbf{J_{train}}$ using transformer-based ASR model, adding the generated non-native accented speech data to fine-tune further degrades the WER performances of the CGN test sets seriously. For the test sets in the Jasmin-CGN corpus, except that the WER of $\mathbf{H_N}$ becomes higher, the WERs of all the other 3 test sets becomes lower, leading to further bias reduction for both read and HMI types speech data.

4.3.2 DAT

DAT with $\mathbf{C_{train}}$ and $\mathbf{J_{train}}$

For both the RNN-based and transformer-based ASR models, the experimental results of DAT training on $\mathbf{C_{train}}$ and $\mathbf{J_{train}}$ are shown in Table 4.1 (with "**DAT**" in the "**Model**" column and $\mathbf{C_{train}}$,$\mathbf{J_{train}}$ in the "**Train data**" column).

For the RNN-based DAT experiment, compared with the experimental results of the standard training method using the same training data setting (for RNN-based ASR model), the main differences are:

- For the CGN test sets, the WER value of $\mathbf{C_{BN}}$ is 5.62% lower, while the WER value of $\mathrm{C_{CTS}}$ is 9.13% higher.
- For all the test sets in the Jasmin-CGN, the WERs all become higher a lot, leading to the obvious increase for both the read and HMI non-nativeness bias.

For the transformer-based DAT experiment, compared with the experimental results of the standard training method using the same training data setting (for transformer-based ASR model), except for $\mathbf{B_H}$, all the evaluation metrics listed in Table 4.1 become worse.

Generally, with training data $\mathbf{C_{train}}$ and $\mathbf{J_{train}}$, the DAT training method does not outperforms the standard training methods for both the RNN-based and transformer-based ASR models.

DAT with $\mathbf{C_{train}}$,$\mathbf{J_{train}}$, $\mathbf{vc_{all}}$, $\mathbf{sp_{all}}$

I conduct the DAT experiments with the transformer-based ASR model, using $\mathbf{J_{train}}$, $\mathbf{C_{train}}$ and all the available generated non-native accented speech data i.e. $\mathbf{vc_{all}}$ and $\mathbf{sp_{all}}$. The corresponding experimental results are shown in Table 4.1 (see the row with both "**DAT: TF-based**" and $\mathbf{C_{train}}$,$\mathbf{J_{train}}$, $\mathbf{vc_{all}}$, $\mathbf{sp_{all}}$). Comparing with DAT experimental results with $\mathbf{C_{train}}$,J_{train} as the "**Train data**", all the evaluation metrics in Table 4.1 become better a lot. Moreover, comparing with the standard training experiment (the same "**Train data**" setting, for the transformer-based ASR model), the DAT training method leads to very similar results.

5 Discussions and Conclusions

5.1 Discussions

Before going into the detailed discussion with the bias mitigation methods applied in this thesis, there are some findings related to bias discovery and quantification in the SOTA E2E ASR models.

I choose 2 SOTA ASR architectures with quite similar speech recognition performance for English in the SpeechBrain toolkit, however, when I use these 2 models to build Dutch ASR systems, they show big performance gap. Beyond my main research question "How to mitigate bias against non-native accents?", I find that the transformer-based ASR model is more suitable than RNN-based one for Dutch speech recognition. Even the two models are with huge WER performance gaps, the bias against non-native accents in the two models are quite similar for both read and HMI types speech.

For the sub-question RQ1: **Are non-native speech data augmentation methods (i.e., speed perturbation, VC-based data augmentation) helpful for bias mitigation against non-native accents in the two SOTA E2E ASR models (i.e., RNN-based and transformer-based ASR models)?**

Both the cross-lingual VC-based and the speed perturbation-based non-native accented speech data augmentation are helpful for non-nativeness bias mitigation in the transformer-based ASR model, but does not contribute de-biasing performance or leads to small bias reduction at the expense of speech recogniton performance for the RNN-based ASR model, indicating that data augmentation methods used in this thesis for bias mitigation is model-dependent. Comparing the two data augmentation methods, speed perturbation is slightly better than cross-lingual VC-based method. For the transformer-based ASR model, both the read and HMI types non-native accented speech data augmentation by the VC method and speed perturbation method have contributions for bias mitigation against non-native accents.

With fine-tuning or DAT training methods, both data augmentation methods are helpful for bias reduction. With the transformer-based ASR model using the standard training method, combining the VC generated and speed perturbed non-native accented speech leads to the best bias mitigation results.

For the sub-question RQ2: **"Do the DAT or fine-tuning have the ability to use the available non-native accented speech data more effectively than the standard training method, resulting in bias reduction?"**

For both RNN-based and transformer-based ASR model, the fine-tuning method have the ability to mitigate the bias against non-native accents, however, at the expense of native accented speech recognition performance. DAT does not work for bias mitigation for the RNN-based ASR model at all. For the transformer-based model, compared with the standard training method, there is no obvious improvement regarding to the bias.

Finally, based on the discussions and all the experimental results given in this thesis, I give the answers to the main research question: **How to mitigate bias against non-native accents?**

For the RNN-based ASR model, comparing all results show that the ideal bias mitigation method is to add the original non-native accented speech data to the training data. Both data augmentation methods do not work for bias mitigation against non-native accents. The fine-tuning training method has the ability to reduce the bias, however, at the expense of the native accented speech recognition performance. Hence fine-tuning is not the ideal bias mitigation method, either. Compared with the standard training, the DAT training method does not lead to further bias reduction.

For the transformer-based ASR model, comparing all results show that the best non-native accent results and overall WER were obtained when using a standard training approach with both VC and speed perturbed data added. The smallest bias for read speech was found for DAT combined with both data augmentation approaches. The smallest bias for HMI speech was observed for fine-tuning with both data augmentation (but at the cost of native performance). Although DAT improved performance, the improvement is smaller than in [29] and [18]. Their amount of accented speech data was however substantially larger than ours.

5.2 Conclusions

In this work, the objective is to reduce bias against non-native accents using augmentation techniques and by exploring alternate training methods. The results showed that both cross-lingual voice conversion based data augmentation and speed perturbation lead to the improvement of non-native accented speech recognition performance and reductions in bias against non-native accents for all training methods with the transformer-based ASR model. A combination of VC and speed perturbed data gave the lowest WER and smallest bias. Comparison of the standard training approach, fine-tuning, and domain adversarial training showed that the standard training approach gave the best results. The best model was trained with the combination of VC and speed perturbed speech with standard training method using the transformer-based ASR model, and reduced the non-native bias for read data from 28.83% to 13.59%, and for HMI data from 29.49% to 12.62% simultaneously. For the RNN-based ASR model, both the data augmentation and effective training methods does not lead to ideal bias mitigation results, which indicates the bias mitigation in the ASR field is model-dependent.

5.3 Future work

Based on what I have done, I propose many ideas from 5 angles which can be explored in the future.

Inspired from the limitations of this thesis, because of the batch size limitation, I removed 43 hours speech data in CGN corpus as for the total recording length. However, recently, the GPUs in TU Delft High Performance Cluster have been improved, so now it is unnecessary to remove longer speech data. Furthermore, as I mentioned in Chapter 3.1.1, in the experiments so far in both [9] and this thesis, the component *e* speech data in CGN corpus is not used at all. Hence it is worth adding the component *e* speech data to conduct experiments. In addition, in this thesis, I only conduct experiments with the E2E ASR model, and it would be better to understand the generalizability of the debiasing methods I proposed if applying the bias mitigation methods to the hybrid ASR models.

For bias mitigation factors, in [9], there are many more factors have bias in the SOTA hybrid ASR systems other than the non-native accent. It is worth applying the methods used in this thesis on other bias mitigation tasks e.g. gender, regional accents, age etc. to see if those methods still work.

For the baselines, I conduct experiments based on only two SOTA E2E ASR model in SpeechBrain toolkit because of time limitation. However, the toolkit is constantly being refined and supplemented. It is quite interesting to train more ASR models, maybe hybrid models or more kinds of E2E models. And then, applying the bias mitigation methods used in this thesis and seeing the feasibility.

For the data augmentation methods:

- **The amount of the generated non-native accented speech data:** I only augmented about 2-fold non-native accented speech data for the speed perturbation method and the cross-lingual VC method. It is worth exploring if other amount of generated data will conduct better performance for bias mitigation or not, especially for cross-lingual VC-based experiments which is lack of information now.

- **Try more VC models:** since I conduct the VC experiments, about 9 months passed. There are more novel VC models proposed. It is interesting to find more available VC models and find and compare their generated speech data pursuing better non-nativeness bias mitigation.

- **Source-target pair selection:** as mentioned in the Chapter 3.3.1, selecting the source-target speaker pair is also possible, as well as using different speaker similarity methods to select the source-target speaker pair.

- **More data augmentation methods:** there are diverse data augmentation methods, but I only use two methods.

For the effective training methods:

- **Try more effective training methods:** include muli-tasking learning and fine-tuning but freeze different layers. I only use supervised-DAT, while it is also possible to use unsupervised-DAT.

- **Try different classifiers in DAT:** I only use a kind of domain classifier (4 linear layer classifiers for Native VS. Non-native accented speech classification). It is worth exploring other accent classifiers in the DAT to persue better bias mitigation performance.
- **Conduct experiments to find the optimal value for λ in DAT experiments:** even though I have conducted 10 experiments to find the better value for λ, I think it is not enough. Because the DAT does not outperform the standard training method. It is worth to conduct more experiments to find the optimal value for bias mitigation.

References

[1] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, “w2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training,” *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 244–250, 2021.

[2] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, “Pushing the limits of semi-supervised learning for automatic speech recognition,” *arXiv preprint arXiv:2010.10504*, 2020.

[3] Q. Xu, A. Baevski, T. Likhomanenko, P. Tomasello, A. Conneau, R. Collobert, G. Synnaeve, and M. Auli, “Self-training and pre-training are complementary for speech recognition,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 3030–3034.

[4] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, pp. 12 449–12 460, 2020.

[5] W. Chan, D. Park, C. Lee, Y. Zhang, Q. Le, and M. Norouzi, “Speechstew: Simply mix all available speech recognition data to train one large neural network,” *arXiv preprint arXiv:2104.02133*, 2021.

[6] D. S. Park, Y. Zhang, Y. Jia, W. Han, C.-C. Chiu, B. Li, Y. Wu, and Q. V. Le, “Improved noisy student training for automatic speech recognition.” in *INTERSPEECH*, 2020, pp. 2817–2812.

[7] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, “Iterative pseudo-labeling for speech recognition,” *arXiv preprint arXiv:2005.09267*, 2020.

[8] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, “Conformer: Convolution-augmented transformer for speech recognition.” in *INTERSPEECH*, 2020, pp. 5036–5040.

[9] S. Feng, O. Kudina, B. M. Halpern, and O. Scharenborg, “Quantifying bias in automatic speech recognition,” *arXiv preprint arXiv:2103.15122*, 2021.

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

[11] T. Kendall and C. Farrington, “The corpus of regional african american language,” *Version*, vol. 6, p. 1, 2018.

[12] J. Poushter, J. Fetterolf, and C. Tamir, “A changing world: Global views on diversity, gender equality, family life and the importance of religion,” *Pew Research Center*, vol. 44, 2019.

[13] T. Han, H. Huang, Z. Yang, and W. Han, “Supervised contrastive learning for accented speech recognition,” *arXiv preprint arXiv:2107.00921*, 2021.

[14] N. Das, S. Bodapati, M. Sunkara, S. Srinivasan, and D. H. Chau, "Best of both worlds: Robust accented speech recognition with adversarial transfer learning," *arXiv preprint arXiv:2103.05834*, 2021.

[15] Y.-C. Chen, Z. Yang, C.-F. Yeh, M. Jain, and M. L. Seltzer, "Aipnet: Generative adversarial pre-training of accent-invariant networks for end-to-end speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6979–6983.

[16] H. Hu, X. Yang, Z. Raeesy, J. Guo, G. Keskin, H. Arsikere, A. Rastrow, A. Stolcke, and R. Maas, "Redat: Accent-invariant representation for end-to-end asr by domain adversarial training with relabeling," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6408–6412.

[17] T. Viglino, P. Motlicek, and M. Cernak, "End-to-end accented speech recognition." in *INTERSPEECH*, 2019, pp. 2140–2144.

[18] H.-J. Na and J.-S. Park, "Accented speech recognition based on end-to-end domain adversarial training of neural networks," *Applied Sciences*, vol. 11, no. 18, p. 8412, 2021.

[19] T. Fukuda, R. Fernandez, A. Rosenberg, S. Thomas, B. Ramabhadran, A. Sorin, and G. Kurata, "Data augmentation improves recognition of foreign accented speech." in *Interspeech*, no. September, 2018, pp. 2409–2413.

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

[21] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *INTERSPEECH*, 2019, pp. 2613–2617.

[22] M. J. F. Gales, A. Ragni, H. AlDamarki, and C. Gautier, "Support vector machines for noise robust asr," in *2009 IEEE Workshop on Automatic Speech Recognition Understanding*, 2009, pp. 205–210.

[23] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[24] Y.-H. Chen, D.-Y. Wu, T.-H. Wu, and H.-y. Lee, "Again-vc: A one-shot voice conversion using activation guidance and adaptive instance normalization," in *ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 5954–5958.

[25] M. Baas and H. Kamper, "Voice conversion can improve asr in very low-resource settings," *arXiv preprint arXiv:2111.02674*, 2021.

[26] S. Shahnawazuddin, N. Adiga, K. Kumar, A. Poddar, and W. Ahmad, “Voice conversion based data augmentation to improve children’s speech recognition in limited data scenario.” in *INTERSPEECH*, 2020, pp. 4382–4386.

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

[28] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, “Libri-light: A benchmark for asr with limited or no supervision,” in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 7669–7673, https://github.com/facebookresearch/libri-light.

[29] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, “Domain adversarial training for accented speech recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4854–4858.

[30] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, “SpeechBrain: A general-purpose speech toolkit,” 2021, arXiv:2106.04624.

[31] D. Pinnow, *Acoustic differences among casual, conversational, and read speech*. Michigan State University, 2014.

[32] M. Nakamura, K. Iwano, and S. Furui, “Differences between acoustic characteristics of spontaneous and read speech and their effects on speech recognition performance,” *Computer Speech & Language*, vol. 22, no. 2, pp. 171–184, 2008.

[33] V. Vielzeuf and G. Antipov, “Are e2e asr models ready for an industrial usage?” *arXiv preprint arXiv:2112.12572*, 2021.

[34] L. Deng and D. Yu, “Deep learning: methods and applications,” *Foundations and trends in signal processing*, vol. 7, no. 3–4, pp. 197–387, 2014.

[35] S. Pouyanfar, S. Sadiq, Y. Yan, H. Tian, Y. Tao, M. P. Reyes, M.-L. Shyu, S.-C. Chen, and S. S. Iyengar, “A survey on deep learning: Algorithms, techniques, and applications,” *ACM Comput. Surv.*, vol. 51, no. 5, sep 2018. [Online]. Available: https://doi.org/10.1145/3234150

[36] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, “Review of deep learning: concepts, cnn architectures, challenges, applications, future directions,” *Journal of Big Data*, vol. 8, 2021.

[37] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to sequence learning with neural networks,” *Advances in neural information processing systems*, vol. 27, 2014.

[38] J. Chung, Çaglar Gülçehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *ArXiv*, vol. abs/1412.3555, 2014.

[39] T. Nakatani, “Improving transformer-based end-to-end speech recognition with connectionist temporal classification and language model integration,” in *Proc. Interspeech 2019*, 2019.

[40] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang *et al.*, “A comparative study on transformer vs rnn in speech applications,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 449–456.

[41] A. Graves, “Supervised sequence labelling,” in *Supervised sequence labelling with recurrent neural networks*. Springer, 2012, pp. 5–13.

[42] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, “Domain-adversarial training of neural networks,” *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[43] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, “Domain adversarial training for accented speech recognition,” 2018. [Online]. Available: https://arxiv.org/abs/1806.02786

[44] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, “State-of-the-art speech recognition with sequence-to-sequence models,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4774–4778.

[45] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, “Conformer: Convolution-augmented Transformer for Speech Recognition,” in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[46] L. Dong, S. Xu, and B. Xu, “Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5884–5888.

[47] M. Sawalha and M. A. Shariah, “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 *Proceedings of the 2nd Workshop of Arabic Corpus Linguistics WACL-2*, 2013.

[48] M. Adda-Decker and L. Lamel, “Do speech recognizers prefer female speakers?” in *Proc. Interspeech 2005*, 2005.

[49] S. Goldwater, D. Jurafsky, and C. D. Manning, “Which words are hard to recognize? prosodic, lexical, and disfluency factors that increase asr error rates,” in *Proc. ACL*, 2008.

[50] A. Koenecke, A. Nam, E. Lake, J. Nudell, M. Quartey, Z. Mengesha, C. Toups, J. R. Rickford, D. Jurafsky, and S. Goel, "Racial disparities in automated speech recognition," *Proceedings of the National Academy of Sciences*, vol. 117, no. 14, pp. 7684–7689, 2020. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.1915768117

[51] Y. Qian, K. Evanini, X. Wang, C. M. Lee, and M. Mulholland, "Bidirectional LSTM-RNN for Improving Automated Assessment of Non-Native Children's Speech," in *Proc. Interspeech 2017*, 2017, pp. 1417–1421.

[52] L. Moro-Velazquez, J. Cho, S. Watanabe, M. A. Hasegawa-Johnson, O. Scharenborg, H. Kim, and N. Dehak, "Study of the Performance of Automatic Speech Recognition Systems in Speakers with Parkinson's Disease," in *Proc. Interspeech 2019*, 2019, pp. 3875–3879.

[53] B. M. Halpern, R. van Son, M. van den Brekel, and O. Scharenborg, "Detecting and analysing spontaneous oral cancer speech in the wild," in *Proc. Interspeech 2020*, 2020.

[54] M. Schuster, A. K. Maier, T. Haderlein, E. Nkenke, U. Wohlleben, F. Rosanowski, U. Eysholdt, and E. Nöth, "Evaluation of speech intelligibility for children with cleft lip and palate by means of automatic speech recognition." *International journal of pediatric otorhinolaryngology*, vol. 70 10, pp. 1741–7, 2006.

[55] Y. Wu, D. Rough, A. Bleakley, J. Edwards, O. Cooney, P. R. Doyle, L. Clark, and B. R. Cowan, *See What I'm Saying? Comparing Intelligent Personal Assistant Use for Native and Non-Native Language Speakers.* New York, NY, USA: Association for Computing Machinery, 2020. [Online]. Available: https://doi.org/10.1145/3379503.3403563

[56] P. Fung and L. W. Kat, "Fast accent identification and accented speech recognition," in *In Proc. ICASSP*, 1999, pp. 221–224.

[57] S. Feng, O. Kudina, B. M. Halpern, and O. Scharenborg, "Quantifying bias in automatic speech recognition," 2021. [Online]. Available: https://arxiv.org/abs/2103.15122

[58] N. Oostdijk *et al.*, "The spoken dutch corpus. overview and first evaluation." in *LREC*. Athens, Greece, 2000, pp. 887–894.

[59] D. A. van Leeuwen, J. M. Kessens, E. Sanders, and H. van den Heuvel, "Results of the n-best 2008 dutch speech recognition evaluation," in *INTERSPEECH*, 2009.

[60] C. Cucchiarini, H. V. Hamme, O. v. Herwijnen, and F. Smits, "Jasmin-cgn: Extension of the spoken dutch corpus with speech of elderly people, children and non-natives in the human-machine interaction modality," 2006.

[61] M. K. Yamagishi Junichi, Veaux Christophe, "CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92)," 2019.

[62] F. Bous, L. Benaroya, N. Obin, and A. Roebel, "Sequence-to-sequence voice conversion using f0 and time conditioning and adversarial learning," *ArXiv*, vol. abs/2110.03744, 2021.

[63] H. Hemati and D. Borth, "Continual speaker adaptation for text-to-speech synthesis," *ArXiv*, vol. abs/2103.14512, 2021.

[64] D. Min, D. B. Lee, E. Yang, and S. J. Hwang, "Meta-stylespeech : Multi-speaker adaptive text-to-speech generation," in *ICML*, 2021.

[65] S. Furui, "Recent advances in spontaneous speech recognition and understanding," in *ISCA & IEEE workshop on spontaneous speech processing and recognition*, 2003.

[66] W. Chou and B.-H. Juang, *Pattern recognition in speech and language processing.* Crc Press, 2003.

[67] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 4879–4883.

6

Appendix

In this Appendix, I mainly describe 3 *contents:*

1. *The detailed WER results on Jasmin-CGN small test sets (mentioned in Chapter 3.2.2),*
2. *The validation of the hypothesis proposed in Chapter 3.4.1, which is beyond the research question in this thesis,*
3. *The exploration of the hyperparameter* λ *in the DAT experiments (mentioned in Chapter 3.4.2).*

6.1 The detailed WER results on Jasmin-CGN small test sets

All the detailed WER results of the 10 small test sets i.e., $\mathbf{R/H_{DC}}$, $\mathbf{R/H_{DT}}$, $\mathbf{R/H_{DOA}}$, $\mathbf{R/H_{NNT}}$ and $\mathbf{R/H_{NNA}}$ in the Jasmin-Corpus are listed in Table 6.1 follow the same description order as the Table 4.1.

For the experimental results shown in Table 6.1 with the same "**Train data**" settings, the WERs on all the 10 small test sets in Jasmin-CGN obtained by the transformer-based ASR model are always lower a lot than those obtained by the RNN-based ASR model, indicating the transformer-based ASR model has better Dutch speech recognition performance then the RNN-based one.

For the same ASR model, compared the experimental results of only using C_{train}, adding $\mathbf{J_{train}}$ to the training data leads to the big performance improvement for the test sets in Jasmin-CGN corpus. Further additions to the VC generated and speed perturbed non-native accented speech data lead to a consistent rise in the speech recognition performance for both the native and non-native accented speech.

Compared with the standard training method, applying fine-tuning always brings benefits for the small test sets in Jasmin-CGN, while applying DAT does not.

Among the 10 small test sets, only for the test set $\mathbf{H_{DC}}$, all the data augmentation methods or the effective training strategies for bias reduction bring negative influence.

Table 6.1: Detailed experimental results on the 10 Jasmin-CGN small test sets. The bold notation means the best performance in each coulumn.

Model	Train data	R_{DC}	R_{DT}	R_{DOA}	R_{NNC}	R_{NNA}	H_{DC}	H_{DT}	H_{DOA}	H_{NNC}	H_{NNA}
BL: RNN-based	C_{train}	45.64	27.98	30.41	61.16	65.21	52.50	40.58	36.13	71.56	67.79
BL: TF-based	C_{train}	35.34	18.35	22.37	50.89	56.40	41.90	25.20	28.61	57.01	61.00
RNN-based	C_{train},J_{train}	16.30	4.70	9.72	33.61	23.49	26.79	20.29	32.90	36.83	43.69
TF-based	C_{train},J_{train}	8.70	3.51	4.32	23.50	18.97	**17.11**	15.12	21.86	29.75	38.02
RNN-based	C_{train},J_{train},vc_{all}	17.54	7.51	10.83	34.46	24.88	28.21	25.86	31.39	39.19	45.41
TF-based	C_{train},J_{train},vc_{all}	8.31	**2.94**	3.84	21.71	16.14	19.45	13.93	22.41	27.92	35.77
TF-based	C_{train},J_{train},vc_{rd}	8.93	3.07	4.42	22.16	16.99	19.87	14.06	22.69	32.11	39.21
TF-based	C_{train},J_{train},vc_{hmi}	9.19	3.31	4.62	22.50	18.57	18.78	**13.40**	22.29	29.10	36.87
RNN-based	C_{train},J_{train},sp_{all}	18.18	6.88	8.50	32.94	23.76	29.97	24.67	32.29	39.84	43.78
TF-based	C_{train},J_{train},sp_{all}	8.48	3.31	3.62	21.40	16.25	19.03	15.12	21.46	**25.16**	34.97
TF-based	C_{train},J_{train},sp_{rd}	8.59	3.07	**3.53**	21.31	16.41	18.03	13.93	22.76	29.23	37.40
TF-based	C_{train},J_{train},sp_{hmi}	8.55	3.31	4.42	23.12	18.42	17.78	15.78	22.03	27.79	35.65
TF-based	C_{train},J_{train},vc_{all},sp_{all}	7.88	3.14	3.75	21.05	**15.86**	18.11	14.99	**20.87**	25.43	**33.78**
F-T: RNN-based	J_{train}	12.39	4.18	8.26	28.91	21.25	24.12	21.09	33.11	36.70	43.07
F-T: TF-based	J_{train}	7.82	3.33	4.23	21.83	19.09	18.36	14.46	23.30	26.34	37.28
F-T: TF-based	J_{train},vc_{all},sp_{all}	**7.69**	3.23	3.80	21.09	17.52	19.45	14.85	26.77	**25.16**	36.03
DAT: RNN-based	C_{train},J_{train}	21.24	14.07	38.28	35.25	16.90	38.48	34.75	54.78	57.73	35.59
DAT: TF-based	C_{train},J_{train}	10.30	3.59	5.01	24.03	19.73	20.37	16.05	23.89	33.03	40.07
DAT: TF-based	C_{train},J_{train},vc_{all},sp_{all}	8.16	2.96	3.90	**21.00**	15.96	19.12	14.19	21.63	26.21	34.37

6.2 Beyond the research question

The hypothesis proposed in Chapter 3.4.1 are:

- When only using $\mathbf{J_{train}}$ as the training data, it is possible to build an ASR system with low bias agasint non-native accents, because the amount of speech data spoken by native speakers and non-native speakers are relatively balanced.

In order to validate the hypothesis, I conduct experiments with the transformer-based ASR model. The experimental results are listed in Table 6.2. When only using $\mathbf{J_{train}}$ to train the ASR model, for the read speech data, the bias $\mathbf{B_R}$ is quite big, while for the HMI speech data, the bias $\mathbf{B_R}$ nearly disappear. Hence my assumption is partially correct.

Without using $\mathbf{C_{train}}$, I conduct more experiments: adding the generated non-native accented speech data ($\mathbf{vc_{all}}$ or $\mathbf{sp_{all}}$ or the both) to the training data, in order to see how the bias change. The experimental results in Table 6.2 shows that for the read speech, the bias $\mathbf{B_R}$ is continuously reduced, while for the HMI speech, the non-native accented speech recognition performance is better than that of the native accented speech.

Table 6.2: The experimental results for the WERs performance and non-nativeness bias quantification. All the numbers with bold notifications means the best value in each column.

Details		**CGN (% WER)**		**Jasmin (% WER and Bias)**					
Model	**Train data**	C_{BN}	C_{CTS}	R_N	R_{NN}	H_N	H_{NN}	B_R	B_H
TF-based	J_{train}	98.23	97.26	30.51	66.08	84.64	85.77	35.57	**0.86**
TF-based	J_{train}, vc_{all}	98.01	96.45	21.23	49.34	81.78	79.70	28.11	-2.08
TF-based	J_{train}, sp_{all}	97.68	95.64	12.86	35.69	71.66	69.05	22.83	-2.61
TF-based	J_{train}, vc_{all}, sp_{all}	**97.22**	**93.19**	**9.19**	**30.18**	**67.20**	**65.76**	**20.99**	-1.44

6.3 DAT hyperparameter exploration

In addition to conducting DAT experiments in Chapter 4.3.2 with $\lambda = 0.01$, I also explore 9 more conditions of the hyperparameter λ so as to find the better setting. I conduct these hyperparameter related experiments with the best training data setting i.e., using $\mathbf{C_{train}}$, $\mathbf{J_{train}}$, $\mathbf{vc_{all}}$ and $\mathbf{sp_{all}}$ and the better ASR model i.e., the transformer-based ASR model according to the experimental results shown in Table 4.1. All the ASR models in this section are trained for 11 epochs.

Table 6.3: DAT hyperparameter exploration experimental results (WERs and bias). The bold notification means the best performance in each column.

Details		**CGN (% WER)**		**Jasmin (% WER and Bias)**					
λ	**Train data**	C_{BN}	C_{CTS}	R_N	R_{NN}	H_N	H_{NN}	B_R	B_H
0.02	–	11.13	39.06	5.40	18.60	21.51	34.15	13.20	12.64
0.03	–	10.16	37.82	5.27	18.35	20.51	33.45	**13.08**	12.94
0.04	–	9.92	37.43	**4.19**	18.45	20.70	**32.69**	14.26	11.99
0.05	–	10.31	37.93	4.82	18.30	21.32	32.87	13.48	**11.55**
0.06	–	**9.44**	37.21	5.00	18.42	20.64	33.47	13.42	12.83
0.07	–	10.51	38.14	5.05	18.39	21.27	33.45	13.34	12.18
0.08	–	9.62	36.75	4.40	**17.88**	20.54	33.23	13.48	12.69
0.09	–	9.92	**37.09**	4.71	18.01	**20.19**	33.10	13.30	12.91
0.1	–	9.97	37.35	4.89	18.44	20.95	33.71	13.55	12.76

The DAT hyperparameter exploration experimental results are listed in Table 6.3. For my experimental settings (training data, transformer-based ASR model), modifying the hyper-parameter λ between 0.01 and 0.1 does not lead to obvious improvement for both the WERs performance and the non-nativeness bias mitigation.