Speech Processing

Automatic Speech Recognition Lab Report

Anshun Zheng

Word Count: 3219

School of Philosophy, Psychology & Language Sciences,
University of Edinburgh

8 December 2022

Contents

List of Figures

List of Tables

1 Introduction

A century ago, a revolution named "office mechanization" was making waves among secretaries in need of taking notes (Juang & Rabiner, 2005, p. 2). This did not bring automatic recorders immediately, but it marked the unfolding of the development of real automatic speech recognition (ASR) technology (p.1). Modern ASR technology has been progressively applied in such areas as "human-computer interaction" and dictation, which is crucial for people unable to type (Jurafsky & Martin, 2009, p. 319). In this report, I will review the theoretical basis underpinning a speaker-dependent digit recogniser, which can also be extended to the general ASR systems that are adapted to individual differences. By conducting several experiments on this speaker-independent ASR system, I endeavored to investigate potential factors that exert influences on ASR recognition accuracy. The toolkit utilised in the report is Cambridge HTK system and data was recorded by students in past few years' speech processing course.

The experiments begin with the impacts of external devices (e.g., microphones) that help record data. The first experiment compared microphones of two qualities (good vs. bad) and probed into influences on the word error rate (WER). The second and third experiments delved deeper into NLP biases such as gender and accent by inspecting the results of models trained on speakers of different genders and accents. The last experiment was designed to figure out the effects of training data size, but on the way, I also compared the effects of training data size with the effects of mismatching (between training data and test data). The report will conclude by giving an overall discussion of the influences of the aforementioned factors, possible improvements, and benefits of using ASR technology.

2 Background

2.1 Data collection and acoustic features

A speaker-dependent speech recogniser takes a waveform signal input to "deduct" the potential word strings. The first step is to collect waveforms of a speaker speaking different digits as training and testing data. Participants were asked to record the following instructions: 1) Repeat every digit (one to ten in random order here) seven times as training data. 2) Another 3 repetitions of all digits for evaluation purposes. 3) Specify the speaker's speech characteristics (i.e., gender, accent). The next step is to label training data with correct transcriptions, including silence (marked as "junk" here), and split test data into isolated digit units. Finally, we need to create a master label file (MLF) containing the golden standard against which the recogniser's performance on test data is compared and evaluated.

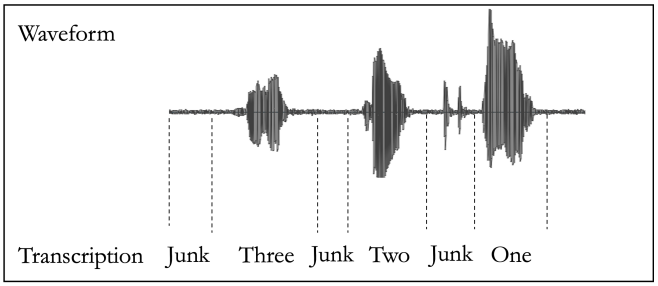

Figure 1: An excerpt of the collected data

Note: there is silence at the edges of each word

Nevertheless, the waveform is not the most appropriate candidate input since it contains too much unnecessary information. Speech is usually considered the result of "source and filter". The filter plays a role in the contour of the spectral envelope, which is helpful in distinguishing phones, while the source involves factors like harmonics, amplitude fluctuation, and gender. Therefore, our goal is to only extract filters.

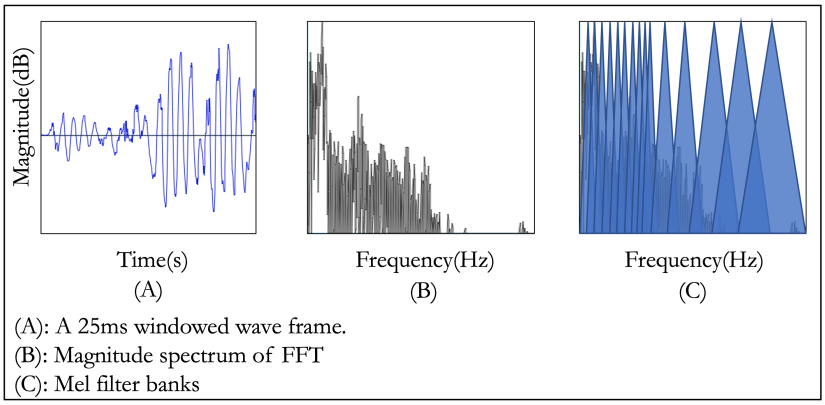

Figure 2: Feature engineering from a waveform to mels

We first preemphasize high-frequency energy and then split the waveform into a series of windows or frames, on which we fit the Fast Fourier transform (FFT). The results of the FFT inform us of the energy distribution among frequency bands (Jurafsky & Martin, 2009, p. 333). However, these filter bands do not reflect the non-linear human auditory perception. Humans are more sensitive to low-frequency changes but insensitive to high-frequency shifts. Moreover, we want to avoid co-variance associated with filter bank outputs before fitting Gaussian models. This is where mel frequency cepstral coefficients (MFCCs) come in.

Basically, we create a filter bank with the first ten filters spaced linearly (less than 1000 Hz) and the rest spread logarithmically (see Figure 2.C). However, the spectrum does not eliminate harmonics. The solution is cepstrum-the spectrum of the log spectrum[1]. We usually take the first few cepstral levels (normally 12), which represent the filter, and then truncate higher values, which are related to Fo (Jurafsky & Martin, 2009, p. 335). The cepstral coefficients plus one energy value are stored in a vector representing the features of one frame. Furthermore, we want to capture the inter-frame change, so we add a delta and a double delta to every current feature, tripling the vector's size.In the mean time, these coefficients are not co-varying which drastically reduces the computation we need. Now we have transformed the waveforms to MFCCs for both training and test data.

[1]Take the log of Mel bank filter outputs and then fit a Discrete Cosine Transform (Gupta et al., 2013, p. 104)

2.2 Training HMMs

The goal of a digit recogniser is given a sequence of feature vectors (observations), to judge what digit it is. We can model this as below:

$$\begin{aligned} W^* = P(W|O) &= \frac{P(O|W) * P(W)}{P(O)} \\ &= \underset{W}{\mathbf{argmax}}\, P(O|W) * P(W) \end{aligned} \tag{1}$$

where:

W^* = The optimal sequence of words; digit here
W = Sequence of words; digit
O = Observations (i.e., feature vectors)

we will start from acoustic model ($P(O|W)$). In another word, we created several digit models that tells us the probability of seeing a sequence of observations given that model ($P(O|model)$) and we label the sequence according to the highest probability. Each digit model here is a Hidden Markov Model (HMM). There are essentially three elements of HMM: 1) Finite states possessing "distinctive" and "measurable" features. 2) A transition to a new state at each time step. 3) An observation emission accompanying that transition (Rabiner & Juang, 1986, p. 7). In our model, the states are a series of Gaussian models (representing different sub-word levels such as phonemes) that "emit" feature vectors because they account for the variability of natural speech and are simple to operate. Furthermore, the Gaussian models we implemented here is Gaussian Mixture Model(GMM) because the feature vectors usually have high dimensions and some dimensions might not obey normal distribution (Jurafsky & Martin, 2009, p. 345).

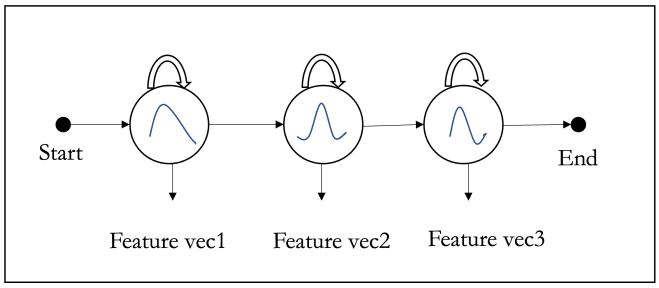

Figure 3: A 5-state HMM for digit recognition

Note: the number of hidden states can vary; three is good enough for the digit recognition task

The second element of HMM, a transition matrix, is quite straightforward. Because speech direction is monotonic (left-to-right topology), the transition is restricted to left-to-right plus the self-loops, which adjust the duration of the same phone. When we train the transition matrix, we initialise it by setting all impossible movements to 0 and giving the same probability to other transitions (i.e., 0.5).

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Figure 4: A transition matrix of a 5-state HMM

Now, how to model the emission matrix emerges as the crux of the issue. Each GMM has two parameters, namely mean(μ) and standard deviation(σ), then the problem becomes how to model $P(O|\mu, \sigma)$. However, when we fit the data to the model, we will find that one state or model can emit a range of feature vectors; that is, we do not know which cluster contributes to the mean and variance of which model (see Figure 5), so the alignment between data and Gaussian models is another nontrivial problem. A cumbersome way is to marginalise and calculate the probability over all possible state sequences, but it is too computation-unfriendly. That is why we turn to the forward-backward algorithm (also known as the Baum-Welch algorithm).

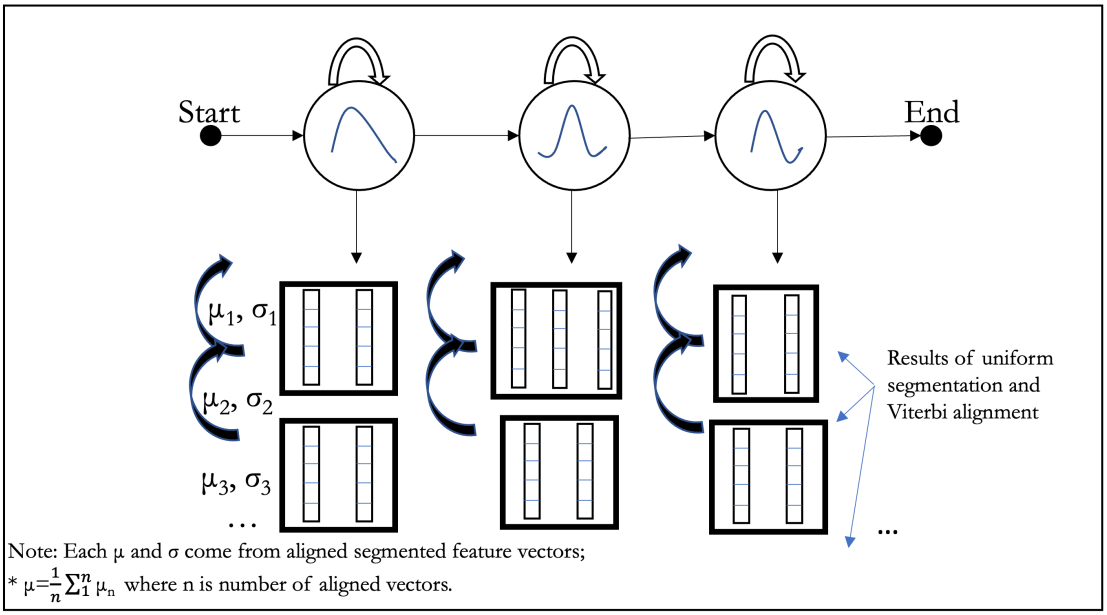

Figure 5: Possible alignments between an observation sequence and states in HMM

Same as transition matrix training, we initialize emission matrix from a "flat start", tuning μ and σ "identically to the global mean and variance of the entire training data". (Jurafsky & Martin, 2009, p. 360).

$$\begin{bmatrix} 0 & 0 & \dots & 0 & 0 \end{bmatrix} \qquad \begin{bmatrix} 1 & 1 & \dots & 1 & 1 \end{bmatrix}$$

(a) B matrix - Mean initialised matrix

(b) B matrix - Variance initialised matrix

Figure 6: Initial values for each GMM state

Both matrices have 39 dimensions

These initialisations will be further revised by "HInit" in HTK and we will obtain the primitive parameters and alignment between observations and states through uniform segmentation and iterative Viterbi alignment[2](Figure 5). After having initial A and B matrices, we run a iteration of Baum-Welch(BW) algorithm to re-estimate the parameters until they converge (Jurafsky & Martin, 2009, p. 361). Figure 7 depicts some intuitions of BW algorithm: Each observation can contribute to different states; in another word, at a given time step t, different states have different emission probabilities,

[2]See Viterbi algorithm in section 2.4

so we want to add weights to those states, which represent the emission probabilities. We can compute the weights or probability of state q_i at time step t_j by adding the probabilities of all previous paths (the forward algorithm) and then combining them with the backward probability to find the global maximum of the weight. HTK provides a handy function-"HRest" to repeat the Baum-Welch algorithm to optimise parameters and store the re-estimated models in another folder. After this step, we will have the optimised Gaussian parameters, transition matrix, and emission matrix.

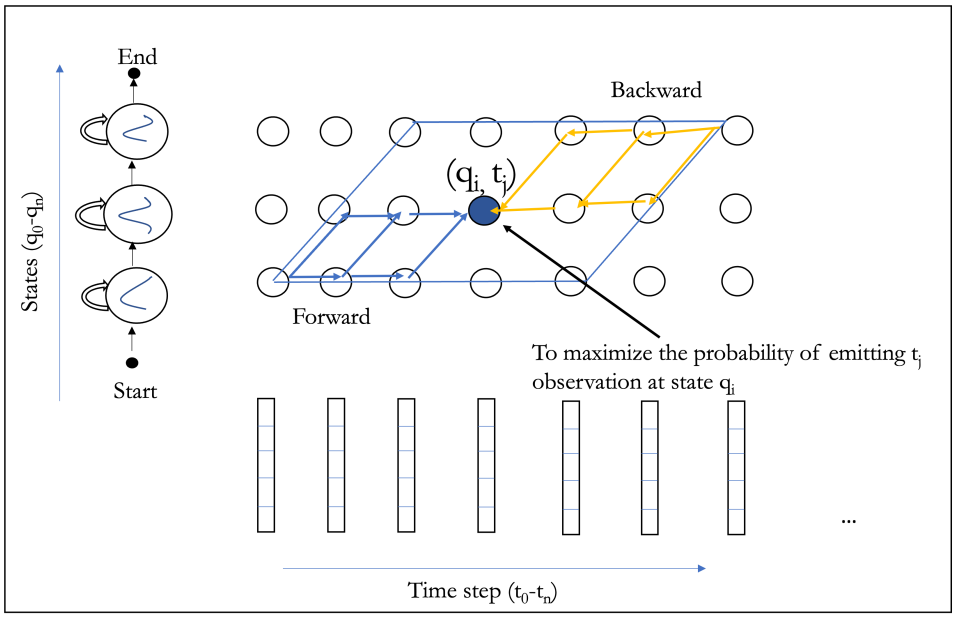

Figure 7: Forward and Backward algorithm intuitions: computing the weight for state q_i at time step t_j

Based on (Simon, 2022, p. 43) and (Bilmes et al., 1998, p. 7)

2.3 Language modelling

Acoustic model has supplied us a sequence of sub-word level (i.e., phonemes and characters) hitherto, whereas this does not guarantee a good word-level sequence. As a result, we include a language model ($P(W)$), which instructs the acoustic model to discard predictions that are grammatically impossible. One commonly used language model is the n-gram model. Take the tri-gram model as an example: it assumes the probability of one word is conditioned on the previous two words ($P(w_n|w_{n-1}w_{n-2})$). For example, sequence like "a cat the" will be excluded since $P(the|a\,cat)$ is almost 0.

Nevertheless, our recogniser does not have a strict language model, instead, we assign a uniform probability to all digits and 0 for all other words. HTK has a function "HParse" to help write grammars, which are compiled into finite state models (Young et al., 2015, p. 212). In our case, we would have a finite state model as in Figure 8, with each transition equally likely to happen so as to ensure the HMM model of every digit is possible to run.

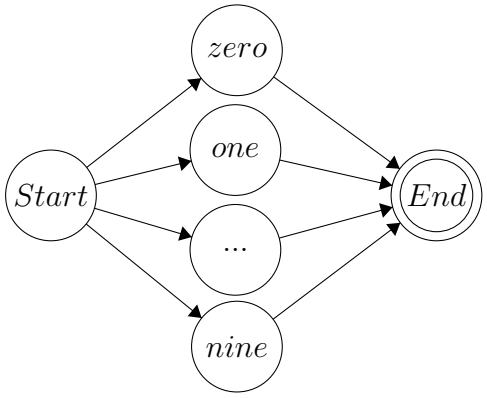

Figure 8: A HKT word network for digit recognition

2.4 Recognition using HMMs

So far, we have trained an acoustic model and a language model, moving on to a stage of decoding. In another words, given some parameters(μ, σ) and an sequence of unlabeled observations, how can the maximum of $P(W|O)$ be computed, that is, $\underset{W}{\mathbf{argmax}}\ P(O|W)*P(W)$ after Bayesian inference. It is inferred from the equation that we need to find an optimal state sequence Q that matches O and maximises the product. Note that the language model here assigns the same probability to each digit and zero probability to other words so it does not interfere too much.

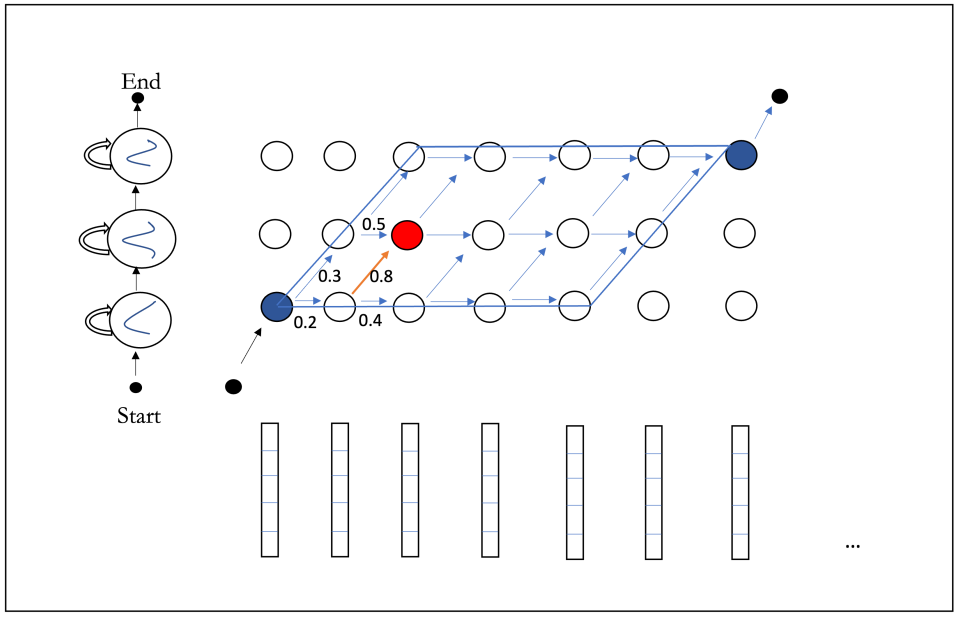

Figure 9: Viterbi algorithm

Note: numbers on the arcs are imaginary emission probabilities, for example, 0.2 means the probability of the first state generating the second observation.

Viterbi is one of many ways to efficiently retrieve the best state sequence. The idea is two-fold: 1) It is dynamic, which means it prunes paths that have a higher cost (lower emission probability here) on the way, so it is fast. 2) It is memoryless: which means that future steps rely solely on the previous one and not on all previous history. Figure 9 roughly illustrates the basic operation of the Viterbi algorithm. It tries to find the path with the highest emission probability from the start state to the end state (marked in blue).

The probability of each feature vector generated by every state will be calculated once at the begin-

ning[3]. Viterbi can start with several parallel paths, but when two paths meet, the one with the lower probability will be eliminated. For example, to reach the state in red, there are two paths with probabilities of 0.15 (0.3*0.5) and 0.16 (0.2*0.6), respectively, and the first path will be pruned. When the algorithm goes further, it will not review the past possible paths. Each HMM will generate one such probability of an optimal path through the "HVite" function in HTK. All digit HMM models will be applied to the given observation sequence and the digit represented by the HMM that generates the highest probability will be selected and labelled to the unknown observation sequence (Figure 10).

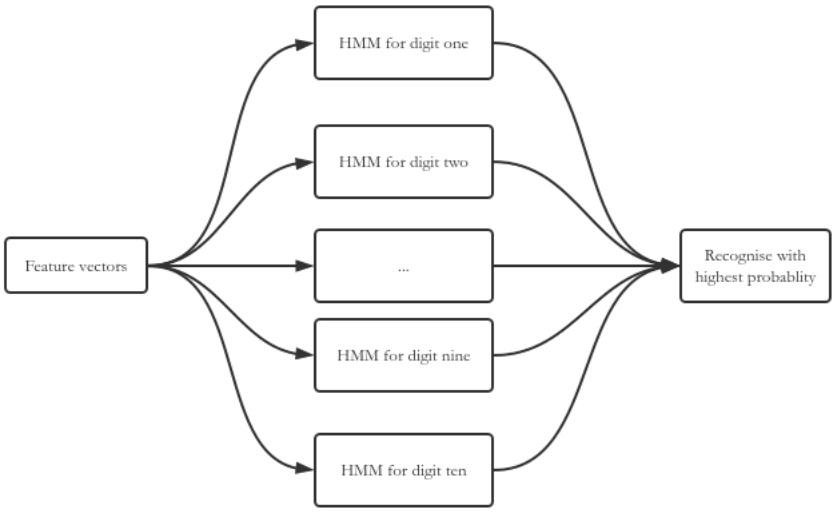

Figure 10: Recognition with HMMs

As soon as the system has the estimated digit for each test file, the "HResults" function will compared the estimated results with the "answer" (mlf file here) to compute the accuracy of the model by $\frac{N(correctly\ identified\ digits)}{N(all\ digits)}$.

[3] This is why it has less computation than the exhaustive search.

3 Experiments

The ASR system involves many variations in either the data set (i.e., in terms of gender, devices, etc.) or algorithms. This section will demonstrate a series of experimental designs along with their results to probe into the effects of various factors.

3.1 Experiment 1: Effects of microphone types

The first experiment investigates how microphone types (small vs. big) influence the WER of the system. Initially, I collected a bunch of data with only the microphone type varying, controlling all other variables (50% male, 50% female; all from non-native speakers, see Table 2). The test data was a mixture of big and small microphone recordings from another set of speakers, with other variables controlled as in training data. One model is trained on the data recorded by a big microphone (group 1: 1-4), while another model is trained on the data recorded by a small microphone (group 2: 5-8). It is important to note that a large microphone has high quality and thus captures a wider range of frequencies. I conjecture that the model trained on big microphones will lead to higher accuracy, because they contain wider range of frequency information to help out detection.

Subject ID	Gender	Microphone type(train)	Accent
1	m	big	NN
2	m	big	NN
3	f	big	NN
4	f	big	NN
5	m	small	NN
6	m	small	NN
7	f	small	NN
8	f	small	NN

Table 1: Details of the selected training data

Subject ID	Gender	Microphone type(test)	Accent
1	m	big	NN
2	m	small	NN
3	f	big	NN
4	f	small	NN

Table 2: Details of the selected test data

As anticipated, the results (see Table 3) turned out that a big microphone will lead to higher accuracy (80.66% vs. 64.09%). It is noticeable that the small microphone group misidentified "one", "two", "four", and "five" more frequently. Possible causes could be lower range of frequency was not captured in "one" and "two" and higher frequencies were not recognised in the rest two words. Another interesting finding was that for most erroneous cases, words were mistaken for "nine" in the group 2.

	Group 1 (big) ACC=80.66%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**15**		2							1
one	1	**11**				2				4
two	2		**18**				1		4	
three				**11**						
four					**9**	10				
five						**17**		1		
six							**18**			
seven	2							**16**		
eight							3		**15**	
nine		1				1				**16**

	Group 2 (small) ACC=64.09%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**18**									
one	1	**4**						2	1	10
two	5		**8**	2		1				2
three				**18**						
four					**6**	8				5
five	2					**9**	3	2	1	1
six	7						**11**			
seven	2						1	**15**		
eight	4		1				2		**9**	2
nine										**18**

Table 3: Result3.1.1

Note: the correct recognitions are underlined and in bold

To further constrain the differences such as individual voice quality, I trained another model on the same speaker and tested on two settings (big vs. small), which is also helpful in exploring the effect of mismatch between training and test sets.

Group	Microphone type (train)	Microphone type (test)
1	small	small
2		big
3	big	big
4		small

Table 4: Within-subject design

Results as shown in Table 5 revealed that in both small and big training setting, the accuracy was high (100%), whereas a drop happened in group 4 (84%) due to mismatch. In theory, the results should all be high due to overfitting. Nonetheless, the under-performance in group 4 revealed the impact of mismatching. Comparing group 2 and group 4, we can make a primary conclusion that when mismatching occurred, the quality of test data matters more than the quality of training data.

	Group1-3 (ACC=100%)									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**5**									
one		**5**								
two			**5**							
three				**5**						
four					**5**					
five						**5**				
six							**5**			
seve								**5**		
eight									**5**	
nine										**5**

	Group 4 (ACC=84%)								
zero	one	two	three	four	five	six	seven	eight	nine
5									
	4	1							
		5							
			5						
				5					
					2		3		
						1	4		
							5		
								5	
									5

Table 5: Result3.1.2: mismatch between test and training setting

3.2 Experiment 2: Effects of gender

3.2.1 Experiment design

I trained two models based on 50 female speakers' recordings (group A) and 50 male speakers' recordings (group B), which were further tested on 1) mixture of 10 female speakers and 10 male speakers 2) 20 female speakers' recordings (note: these testing sets do not overlap with the training data). All other conditions (i.e., microphone and accents) are controlled in the same proportion (shown in Table 6). I hypothesized that group A would achieve higher accuracy in both test settings and this will be more obvious in setting 2 due to matching gender.

	Proportion
Microphone type	iMac (50%)
	Logitech headset(50%)
Accent	American (8%)
	Non-Native(58%)
	Scottish (8%)
	Other British(26%)

Table 6: Proportions of different accents and microphone types

3.2.2 Results

	Group A (Female) ACC=83.42%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**53**						2			2
one	2	**52**			1		1			
two	9	2	**39**		2	1	1	2		1
three	5			**40**			2	8		
four	4				**53**					
five	1	4			2	**45**	2		1	1
six	1			1			**53**	2		
seven	4		5				1	**52**		
eight				1	2	1	6		**42**	2
nine	1	9				1				**44**

	Group B (Male) ACC=83.07%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**44**	3						2		8
one	1	**52**								3
two	14	6	**35**			1		1		
three	1	4	1	**48**				1		
four		4			**47**	6				
five		1				**46**	1		1	7
six	2						**54**	1		
seven	1	4					1	**50**		1
eight		1	2	3	2	3			**42**	4
nine	1	3								**53**

Table 7: Result3.2.1: gender influences testing on setting 1

	Group A (Female) ACC=86.77%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**49**		2		1		3			1
one	3	**48**			1	1	1			2
two	6		**46**		1	1	1	2		
three	1	1	3	**43**			2	5	0	1
four	3	2			**50**	1		1		
five	1	2				**48**	2	1	1	1
six	1		1			1	**54**			
seven	1				1			**54**	1	
eight		1				1	3		**52**	
nine	1	5			2			1		**48**

Group B(Male) ACC=72.31%									
zero	one	two	three	four	five	six	seven	eight	nine
31	8	2		2	2	1	2		9
1	**46**								9
10	7	**33**	1	1	2		3		
1	10	2	**39**			1	2		1
1	6			**38**	10		1		1
	1				**43**	1	1	1	9
1		1		1		**52**	1		1
1	8			1	1	3	**42**	1	
	6		1	6	3			**38**	3
1	7				1				**48**

Table 8: Result3.2.2: gender influences testing on setting 2

Table 7 has shown similar results on both female and male models, which is different from my hypothesis, and indicated that gender may not influence recognition accuracy in general. However, results in Table 8 revealed that the system trained on female speakers tend to perform better than that trained on male speakers (86.77% vs. 72.31%). The latter was found especially bad at recognising "zero", "four" and "eight" with a accuracy of 54.4%, 66.7% and 66.7% respectively. One reason that group A outperformed group B is that the test set is more similar to the training date in the female group. Sokolov and Savchenko (2021, p. 413) pointed out that female speakers are more likely to have high-frequency consonant noise and vowel formants. Therefore, two models are trained on different distributions of energy due to gender, and the energy distribution of the test set resembles that of the training set of the female model more. Sokolov and Savchenko (2021, p. 413) mentioned that one of the gender adaptive techniques is to select the right acoustic model from multiple AMs for different genders after classifying the "domain" gender by utilising video modality.

3.2.3 Discussion

The results presented above indicated that systems trained on male voices performed worse on female voices, while most real-world ASR systems were trained on male voices. Many ASR systems in various fields including medical, entertainment (Rodger & Pendharkar, 2004; Tatman, 2017; Tatman & Kasten, 2017) have such gender bias. For example, in Tatman (2017), she found out that the word error rate of YouTube captions for female speakers was far higher than for male speakers, and even manipulating the pitch will not offset this difference. In her following study (Tatman & Kasten, 2017), she proved that the gender bias was not unique to YouTube but also existed in Bing speech. There are a variety of ways to mitigate this bias, such as by developing a more varied database and constructing a more flexible measure evaluator (Ngueajio & Washington, 2022, p. 16). To add to that, Markl and McNulty (2022, p. 8) argued that the lack of motivation for companies to reduce predictive bias in

ASR systems appears to be a major barrier, so giving incentives to corporations is important.

3.3 Experiment3: Effects of accent

As with gender bias in the second experiment, accent or race bias is another big issue in NLP. This part is trying to figure out whether models trained on UK English will have high accuracy in recognising other varieties of English. I assume they will be accurate in more similar varieties with UK English and less so in other varieties.

3.3.1 Experiment design and results

I trained a model on 40 UK English speakers, where the proportion of each gender was 50%. The test sets were divided into two sets, one from 20 speakers of different native English varieties (20% Scottish English, 20% Australian English, 60% American English), and another from 20 non-native English speakers. Note that the ratio of men to women was controlled as well.

	Group 1 (Other varieties of English) ACC=82.57%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**44**					1		1		8
one	1	**39**					1			14
two	16		**35**					3		
three			1	**54**						
four	4	8			**26**	14		2		1
five						**52**	1			1
six							**53**	2		
seven	2							**50**		2
eigh				1			6		**47**	
nine	3			1					1	**50**

Group 2 (Non-native English) ACC=74.67%									
zero	one	two	three	four	five	six	seven	eight	nine
51	2					1	1		5
	51					1	5		3
29		**22**				2	7		
3			**46**			4	6	1	
4	15			**30**	3	3	5		
	4			1	**47**	1	4		3
4						**54**	2		
9	1						**50**		
2		3	2			8	1	**44**	
	3			0			3		**53**

Table 9: Result3.3

As expected, the trained model on UK English speakers performed well on those more similar varieties such as American English and Scottish English while the accuracy decreased more around ten percent in Non-native English scenario. This resembles the real world case to some extent since most ASR systems are trained on standard English varieties while harming the less similar varieties (i.e., AAVE).

3.4 Experiment 4: Effects of training data size

Most people, even scientists, would hold the opinion that more data would lead to better and more promising results. This idea was also supported by many studies; for instance, Banko and Brill (2001) conducted a language disambiguation task and a model trained on 1000 times more data performed better, and he said they decreased the error rate simply by adding more data. Nonetheless some studies discovered quite opposite-an inverse relationship between good results and data size (Shalev-Shwartz & Srebro, 2008; Tsangaratos & Ilia, 2016). Some studies claimed that efficient use of data mattered rather than simply expanding data size, so they suggested creating models making good use of big data (Zhu et al., 2016, pp. 76, 91). In my experiment, I suppose that larger data size will lead to higher accuracy.

3.4.1 Experiment design

This experiment was designed to investigate the impact of training data size and compare it with the effect of matching between test and training set, so I trained two models based on respectively 50 speakers and 100 speakers, in which the gender proportion was balanced (50%, 50%). One test group followed the gender proportion, while another did not. There were basically four experiments, as follows:

Group	Training set	Test set
1	50 speakers (50% F, 50% M)	20 speaker (50% F, 50% M)
2	100 speakers (50% F, 50% M)	20 speaker (50% F, 50% M)
3	50 speakers (50% F, 50% M)	20 speaker (70% F, 30% M)
4	100 speakers (50% F, 50% M)	20 speaker (70% F, 30% M)

Table 10: Experimental design: exploring the interaction between size and matching

3.4.2 Results

	Group 1 (50+match) ACC=70.94%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**44**	2	3			3	6	1		1
one	2	**49**	1		1		2	2	5	
two	4		**39**	2	1	0	5		13	
three	2		1	**40**			8		11	
four		10	2		**48**		3	1		
five						**44**	12		3	2
six	1		1				**59**		1	
seven			8			3	5	**45**		
eight							14		**48**	
nine		4		1		1	6	8	3	**38**

	Group 2 (100+match) ACC=78.59%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**52**		3			3	2	1		
one	2	**47**	2		2		2	2	5	
two	3		**48**	9	1		2		1	
three	2		1	**56**			3			
four		3	3		**53**		2	3		
five						**52**	7			2
six	1		3				**57**		1	
seven			11			4	10	**36**		
eight			1				8		**53**	
nine				2		1	4	3	2	**49**

Table 11: Result3.4.1

Table 11 shows that when the training data size doubled, the accuracy increased significantly by almost ten percent. In the meantime, if the matching setting was shifted from mismatch to match, there was a slight increase of about 3 percent (see Table 12 left). The result of group 4 indicates that even if the mismatching reduced the overall accuracy but did not offset the increase brought by data size.

	Group 3 (50+mismatch) ACC=67.5%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**41**	3	2			3	8	2	1	1
one	2	**50**	1		1		2	3	3	
two	9	1	**32**		1	0	8	2	11	
three	2		1	**35**			10	3	11	
four	1	12	2		**39**	1	3	6		
five						**44**	11	1	3	2
six	2		1				**58**		1	
seven			3			3	4	**51**		
eight							12	1	**49**	
nine						1	6	9	3	**33**

	Group 4 (100+mismatch) ACC=79.52%									
	zero	one	two	three	four	five	six	seven	eight	nine
zero	**51**	1	2			3	3	1		
one	2	**48**	2		2		2	3	3	
two	8		**42**	6	1		5	1	1	
three	2		1	**54**			4	1		
four	1	5	3		**48**		1	6		
five						**52**	7			2
six	2						**59**		1	
seven			3			4	11	**43**		
eight							9	1	**52**	
nine				2		1	4	4	2	**48**

4 Discussion and conclusion

This report has so far presented an extensive analysis of the factors that influence the ASR performance. We can draw the following conclusions from the preceding experiments: 1) Microphones of higher quality will improve the WER by capturing more complete frequency information. 2) Models trained on male speakers are likely to perform worse on female voices; in the meantime, models trained on UK English had a lower WER in non-native English recognition. These scenarios mirror the real-world biases that need to be eliminated. 3) More training in our task led to better performance, and its effects outweighed the effects of gender mismatching. However, it may not apply in other tasks, where some research has found a negative relationship.

In a nutshell, ASR technology has been changing the face of the current human-machine interaction industry and is still progressively moving forward. It has provided much convenience to people from all walks of life, so that we no longer have situations like the office revolution that happened a century ago. However, like many other NLP technologies, there are various biases in ASR's database. This issue should be prioritised before developing more "intelligent" models. Otherwise, we may continue to exacerbate the plight of minority groups.

Bibliography

Banko, M., & Brill, E. (2001). Mitigating the paucity-of-data problem: Exploring the effect of training corpus size on classifier performance for natural language processing. *Proceedings of the first international conference on Human language technology research*.

Bilmes, J. A., et al. (1998). A gentle tutorial of the em algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International computer science institute*, *4*(510), 126.

Gupta, S., Jaafar, J., Ahmad, W. W., & Bansal, A. (2013). Feature extraction using mfcc. *Signal & Image Processing: An International Journal*, *4*(4), 101–108.

Juang, B.-H., & Rabiner, L. R. (2005). Automatic speech recognition–a brief history of the technology development. *Georgia Institute of Technology. Atlanta Rutgers University and the University of California. Santa Barbara*, *1*, 67.

Jurafsky, D., & Martin, J. H. (2009). *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (2nd edition)*. Prentice-Hall, Inc.

Markl, N., & McNulty, S. J. (2022). Language technology practitioners as language managers: Arbitrating data bias and predictive bias in asr. *arXiv preprint arXiv:2202.12603*.

Ngueajio, M. K., & Washington, G. (2022). Hey asr system! why aren't you more inclusive? automatic speech recognition systems' bias and proposed bias mitigation techniques. a literature review. *arXiv preprint arXiv:2211.09511*.

Rabiner, L., & Juang, B. (1986). An introduction to hidden markov models. *ieee assp magazine*, *3*(1), 4–16.

Rodger, J. A., & Pendharkar, P. C. (2004). A field study of the impact of gender and user's technical experience on the performance of voice-activated medical tracking application. *International Journal of Human-Computer Studies*, *60*(5-6), 529–544.

Shalev-Shwartz, S., & Srebro, N. (2008). Svm optimization: Inverse dependence on training set size. *Proceedings of the 25th international conference on Machine learning*, 928–935.

Simon, K. (2022). *Speech processing module 10* [University Lecture], University of Edinburgh.

Sokolov, A., & Savchenko, A. V. (2021). Gender domain adaptation for automatic speech recognition. *2021 IEEE 19th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 000413–000418.

Tatman, R. (2017). Gender and dialect bias in youtube's automatic captions. *Proceedings of the first ACL workshop on ethics in natural language processing*, 53–59.

Tatman, R., & Kasten, C. (2017). Effects of talker dialect, gender & race on accuracy of bing speech and youtube automatic captions. *Interspeech*, 934–938.

Tsangaratos, P., & Ilia, I. (2016). Comparison of a logistic regression and naıve bayes classifier in landslide susceptibility assessments: The influence of models complexity and training dataset size. *Catena*, *145*, 164–179.

Young, S. J., Kershaw, D., Odell, J., Ollason, D., Valtchev, V., & Woodland, P. (2015). *The HTK Book Version 3.5*. Cambridge University Press.

Zhu, X., Vondrick, C., Fowlkes, C. C., & Ramanan, D. (2016). Do we need more training data? *International Journal of Computer Vision*, *119*(1), 76–92.