Gender Bias in ASR for Cinema Ticket Reservation

Author
Clara Rus
Data Science Master
S1064211

March 2022

Introduction

Speech recognition can be used to automate or simplify tasks in a variety of domains by taking as input the user's voice. Such applications can be used to help people who are busy with other tasks and even help people with certain disabilities. Speech recognition creates the opportunity for the user to communicate easier with the computer. It can help retrieve or search for certain information, or even execute commands [18].

Some systems can communicate with the user both using text and sound. Such systems are called interactive voice applications (IVR). An IVR uses speech recognition to interpret the user's voice as input and responds using text to speech to create the illusion of a conversation. IVRs are widely used to automate simple tasks such as calls to a call-center, making reservations and appointments etc. [5].

IVRs can be used to automate the reservation of cinema tickets. The user communicates the request using the voice, and then the speech recognition engine interprets the voice and converts it to a string. Next, this string is used to call the function that executes the requested command [15]. It could be the case that a person does not know how to use the Internet to make a reservation for a movie or buy a ticket, or as mentioned before, that person might have a disability, or not see well enough to use a computer. Using an IVR system a person can use their voice to make a reservation for a movie. Thus, such a system might ease this task for certain people while automating the job of cinema workers.

This project will focus on creating an IVR using VoiceXML for a cinema ticket reservation system. The user will call a certain phone number to make the reservation, but instead of communicating with a person, the user will communicate with the IVR using his/her voice. It will first let the user choose the location of the cinema, then a movie from the list of movies that are currently running in the already selected cinema. Next, the user will be asked which is the preferred day and time to come to the cinema for the movie. Finally, the system will ask for the user's details, such as their name to confirm the reservation.

The system will be tested against the presence of gender bias. The presence of bias can lead to discrimination against a certain group as was the case with the Amazon recruitment system [2] which turned out to be favouring the male group over the female group. Another well-known example is the Microsoft Twitter bot [12] that learned within a day to be very racist. In the domain of ASR, authors of [10] looked into the racial bias of commercial speech recognition applications and found that on average the word error rate is higher for recordings of black people than of white people. Several researchers [11] [13] [16] [3] [6] looked into the presence of gender bias in speech recognition systems. Results showed that some ASR systems might show a performance gap between male voices and female voices. As in the case with a medical speech recognition system, this implies that it is harder for female doctors to do their job than it is for male doctors [13]. Car systems that use speech recognition proved to be less performant on female voices and on people speaking with foreign accents [11]. This could lead to possible accidents as the driver might get angry that the system does not listen to the requested command and thus, lose focus on the road while trying to make the system work. Even commercial systems like YouTube's Automatic Captions proved to discriminate against female content creators [16]. This again makes the job of female content creators harder, as they might need to put an extra effort to correct the captions. This also discriminates against the deaf community who rely on correct captions to understand the content of such videos.

Considering all of the above, it is very important to make sure that an ASR system designed to ease a certain task does not discriminate against gender or race. Thus, the voice cinema ticket reservation system will be tested on both male and female voices to make sure that the system performs equally on both groups. This

research will focus on answering the following research questions:
Q1: Does the ASR system for cinema ticket reservation discriminate against gender?
Q2: Does it perform better when using male voices than female voices or vice-versa?

Method

An Interactive Voice Application using VoiceXML follows the structure described in Figure 1. First, the user calls the phone number assigned for this application. Next, the VoiceXML Gateway interprets the user's voice input using Automatic Speech Recognition (ASR), and then it makes a request to the web server. The web server communicates with the database and retrieves or stores the needed information. Then the server responds using a VoiceXML file that the VoiceXML Gateway knows how to interpret and communicates the information back to the user using Text to Speech (TTS).

A VoiceXML gateway is the server that handles the interactions between user's calls and the voice applications defined on a web server. It is equipped with both a telephone card and a connection to the Internet. The interaction and dialogue between the user and the system are defined in the VoiceXML file. The VoiceXML Gateway also provides the speech processing capabilities such as Automatic Speech Recognition and Text to Speech [14]. VoiceXML is the standard markup language used for voice applications. In the same way, as a web server knows how to interpret an HTML file and display a web page, a VoiceXML gateway knows how to interpret the VoiceXML file and receive as input the user's voice while outputting audio information [1].

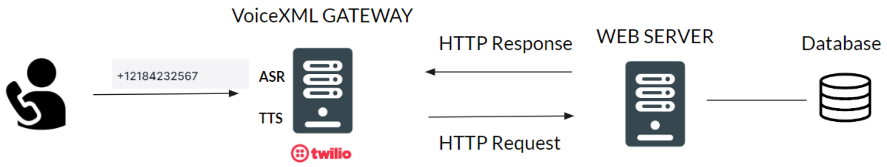
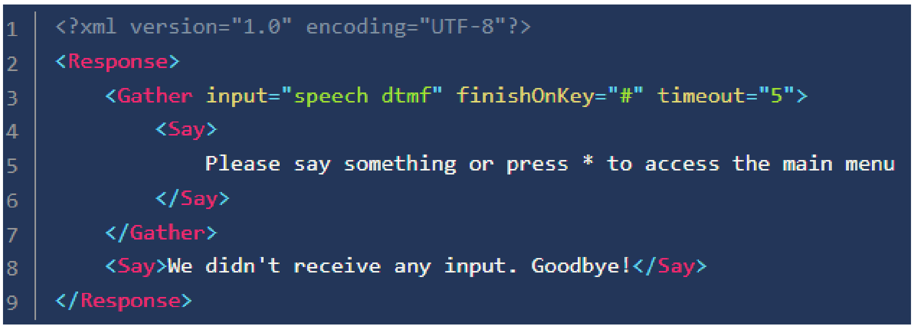


Figure 1: Pipeline of the Cinema Ticket Reservation System

Twilio offers such a service on a free account trial. It offers a US phone number and the possibility to link the voice application to the offered phone number. Twilio works with a proprietary VoiceXML called TwiML, meaning that TwiML can be used only with Twilio phone numbers and gateway. TwiML, the Twilio Markup Language, is used to create the XML document with special tags that define the set of instructions that tell the Twilio gateway what to do when receiving an incoming call, SMS, or fax [17]. It can decide if the voice application is now listening to the user's input or if it will communicate information back to the user using TTS.



```
<?xml version="1.0" encoding="UTF-8"?>
<Response>
    <Gather input="speech dtmf" finishOnKey="#" timeout="5">
        <Say>
            Please say something or press * to access the main menu
        </Say>
    </Gather>
    <Say>We didn't receive any input. Goodbye!</Say>
</Response>
```

Figure 2: Example of TwiML file.

Figure 2 shows an example of a TwiML file that can be used in a voice application. All instructions need to be nested inside the $< Response >$ tag, which is the root element of Twilio's XML markup. Inside this tag, one can define the set of instructions to be executed. Using the $< Say >$ tag the gateway is instructed to communicate the information between the tags to the user using TTS. The $< Gather >$ tags are used to instruct the application that it is supposed to wait for a number of seconds for the user's input. Using the *timeout* attribute one can set the number of seconds to wait for the user's input. If the application does not receive any input it will move to the next instruction, which is a $< Say >$ tag that informs the user about the error. The user can communicate with the application either by voice if the *input* attribute is set to *speech*, or by pressing the keyboard if the *input* is set to $dtmf$. In the scenario from the example, both ways of communication are accepted. Several other useful attributes can be set for the $< Gather >$ tag, such as $finishOnKey$ to inform the system that it does not need to listen to the input anymore if a certain key was pressed. The *language* or the $speechModel$ attributes can be used to increase the accuracy of the ASR. If for example only short commands are expected from the user, then setting this attribute to *numbers_and_commands* should increase the accuracy of the ASR. To instruct the application on what to do after it gathered the input from the user, the *action* attribute is set to a certain URL that provides another set of instructions in the form of an XML file.

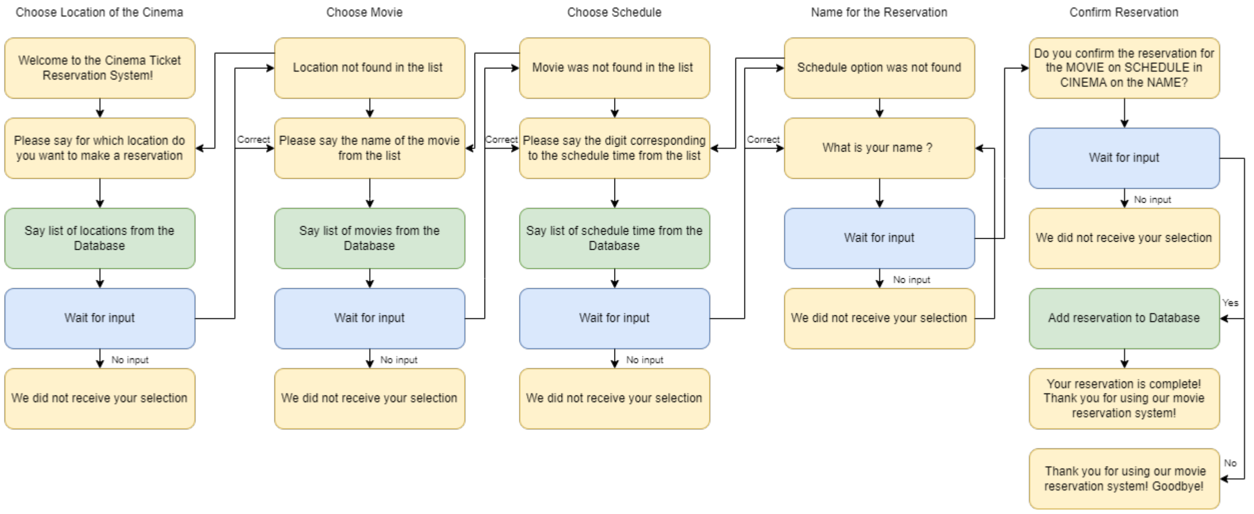

Choose Location of the Cinema
Welcome to the Cinema Ticket Reservation System!
Please say for which location do you want to make a reservation
Say list of locations from the Database
Wait for input
No input
We did not receive your selection
Correct
Choose Movie
Location not found in the list
Please say the name of the movie from the list
Say list of movies from the Database
Wait for input
No input
We did not receive your selection
Correct
Choose Schedule
Movie was not found in the list
Please say the digit corresponding to the schedule time from the list
Say list of schedule time from the Database
Wait for input
No input
We did not receive your selection
Correct
Name for the Reservation
Schedule option was not found
What is your name ?
Wait for input
No input
We did not receive your selection
Confirm Reservation
Do you confirm the reservation for the MOVIE on SCHEDULE in CINEMA on the NAME?
Wait for input
No input
We did not receive your selection
Yes
Add reservation to Database
Your reservation is complete! Thank you for using our movie reservation system!
No
Thank you for using our movie reservation system! Goodbye!

Figure 3: Dialogue for the Cinema Ticket Reservation System.

Figure 3 describes the dialogue that the Cinema Ticket Reservation System will follow. The voice application will receive a set of instructions using the above-mentioned tags, $< Gather >$ and $< Say >$, that will follow the dialogue described in Figure 3. The dialogue is split into five main parts: Choose Cinema's Location, Choose Movie, Choose Schedule, Name for the Reservation and Confirm Reservation. The yellow blocks represent $< Say >$ instructions, the blue blocks represent $< Gather >$ instructions, and the green blocks represent the interaction with the database. If the application does not receive any voice input from the user, it will communicate that to the user and then repeat the message. If the user provides voice input, the ASR will use Speech to Text to transform the voice into a string. If this string is present in the database, the dialogue will move to the next part, otherwise, it will ask the user to choose again something present in the database. The code used to create the Cinema Ticket Reservation System can be found on the following GitHub link: `https://github.com/ClaraRus/ASR_Cinema_Ticket_Reservation`.

Experimental Setup

After the Cinema Ticket Reservation System was created, I tested the Speech to Text functionality against gender bias. For this, I used a set of male and female recordings and evaluated the performance of the ASR using the accuracy and the Word Error Rate (WER). The discrimination is measured using the formula proposed by [3] as inspiration: $disc = 100\frac{P_{male}-P_{female}}{P_{male}+P_{female}}$. P_{male} represents the performance of the male group, while P_{female} represents the performance of the female group. The performance can be either the accuracy or the WER, or any other performance metric. If this discrimination factor is positive, it means that there is a bias towards the male group, otherwise, if the result is negative there is a bias towards the female group. If the result is very close to zero, this means that there is no bias present.

The experiments will be conducted on two datasets: the Google Commands Dataset [9] and the Movie Names dataset. Considering the dialogue above, the system needs to recognize short commands such as: "Yes", "No", digits and movie names. Thus, the above-mentioned datasets are very suitable for testing both the presence of gender bias and the overall performance of the system.

Twilio offers the possibility to choose an ASR model more suitable for short commands and digits. Thus, experiments are conducted using both the "default" and the "numbers and commands" speech models. According to the documentation provided by Twilio [17], the "numbers and commands" speech model is best suited for the use cases where you'd expect to receive short queries such as voice commands or voice searches. In this section, I will refer to the ”default” model as the Default model and to the ”numbers and commands” model, as the Commands model.

To test the consistency of the system, each experiment was conducted three times and the reported results are computed as the average over the repetitions. For each command, from the Google Commands Dataset, both the female and male audios were played three times. In the same manner experiments on the Movie Title dataset were conducted, for each movie, all the files were played three times.

Google Commands Dataset

The Speech Commands Dataset [9] has 65,000 one-second long audios of 30 short words such as: ”Yes”, ”No”, ”Left”, ”Right”, ”One”, ”Two”, etc. spoken by thousands of people. It can be the case that a command was repeated by the same person several times. The data of the speakers is anonymized. Each speaker has an ID, and each audio file that contains the voice of that person has the corresponding ID in the filename.
Considering the limitations of the free trial for a Twilio phone number, the experiments were conducted only against the following commands: ”Yes”, ”No”, and the digit ”One”. As there is no information regarding the speakers of the audio files, I annotated the gender for each of the first 100 samples of each command. However, experiments were performed on an equal number of male audio samples and female audio samples.

Table 1 shows the number of samples and the number of speakers for both genders. In the annotation process, I encountered some difficulties with assessing the gender of the speaker for some of the audio files. Thus, to assess the gender of the speaker, I extracted one audio file corresponding to each speaker and created a form in which I asked the user to assess the gender of the speaker. I distributed the form online and annotated the audio files based on the responses. If the majority of the responses were dominated by one of the genders, then the audio file was used in the experiments, otherwise, the audio file was dropped from the dataset.

Data	Female Speakers	Male Speakers	Samples
Yes	11	8	20
No	9	6	15
One	13	11	25

Table 1: Google Commands Dataset

Movie Titles Dataset

To test the system on movie titles, I created a dataset composed of audio files that contain a person’s voice saying a movie title. In each audio file, a person repeats four times the name of a movie. The audio files contain the voices of three male and three female speakers. In the end, the system is tested on 180 audio samples for each gender group. The movie titles were chosen with different degrees of difficulty, containing one word or more words. Table 2 shows the movie titles used in the experiments.

1 word	2 words	3 words	3+ words
Inception, Encanto, Frozen, Goodfellas, Interstellar	Star Wars, Pulp Fiction, Fight Club, Forrest Gump	City of God, The Dark Knight, Eat-Pray-Love	The Lord of the Rings, Indiana Jones and the Raiders of the Lost Ark, Once Upon a Time in America

Table 2: Movie Titles Dataset

Experiments and Results

Google Commands Dataset

Table 3 shows the results on the three commands extracted from the Google Commands Dataset, using the Default model. For both the female and male groups the accuracy is reported and then based on the accuracy and the formula described in the Experimental Setup Section, the discrimination is calculated.

Looking at the results conducted with the Default model, for all commands it seems that the model favours the male voices. Due to the constraints of the Twilio free account, no experiments were conducted with the Commands model. The highest accuracy for the male audios was obtained on the ”One” command, while the highest accuracy for the female audios was obtained on the ”Yes” command. Computing the average over the

Data	Female	Male	Discrimination
Yes	60%	75%	11%
No	53%	64%	9%
One	46%	76%	24%

Table 3: Results on the Google Commands data using the Default model.

three commands, the accuracy on the female audios is 53% while the accuracy on the male audios is 71%. This results in a discrimination factor of 14% on this dataset.

Table 4 shows the substitution mistakes performed by the "default" model. For the "Yes" command it sometimes made a substitution with "Yeah", which could be considered a correct answer given that it has the same meaning as "Yes". Looking at substitution mistakes for the "No" command, it seems that similar words in pronunciation such as: "Snow", "Now", and "Know" were detected by the ASR model. In addition, it mistaken "No" with "Hello" and "Loan", which are less similar to the given command. For the "One" command, the words which are quite similar in pronunciation are "Why", "When", and "Wanted". In addition, other words less similar were confused with the command such as: "Glen", "Good Morning", and "Phlegm". In addition, by default, the model has profanity censoring, as it seems it got a mistake which was censored: "p***".

	Yes	No	One
Male Audios	Yeah	Snow, Hello, You know	p***, Why?, Wanted Good Morning
Female Audios	Yeah	Snow, Loan, Hello, You know, Now	Why?, When?, Phlegm Glen, Good Morning

Table 4: Mistakes on audios of the Google Commands data

Movie Titles Dataset

Table 5 shows results obtained with the Default model and Table 6 shows results obtained with the Commands model. For movies with titles longer than one word, the WER was also reported for both genders together with the discrimination factor. The results are split into levels of difficulty based on the number of words the titles are composed of. Thus, the results represent the average accuracy and WER over the movie titles in each category. More detailed results can be seen in Figure 4, which shows the accuracy of each movie title separately.

(a) Accuracy

Words	Female	Male	Discrimination
1	18%	38%	35%
2	14%	29%	34%
3	22%	27%	10%
3+	44%	74%	25%

(b) Word Error Rate

Words	Female	Male	Discrimination
2	1.06	0.58	29%
3	0.78	0.54	18%
3+	0.49	0.14	55%

Table 5: Results on the Movie Titles dataset using Default Model

Looking at the results obtained with the Default model, the model seems to favour the male voices over the female voices. Both in terms of accuracy and in terms of WER, the performance of the model is better on the male voices. Averaging the results, the accuracy on the female voices is 32% while the accuracy on the male voices is 42%, resulting in an average discrimination of 13% on the movie titles dataset in terms of the reported accuracy. The word error rate is on average almost two times higher on the female audios, resulting in a discrimination factor of 29%. The highest accuracy for both genders was achieved on the longer movie titles which are composed of more than three words. The lowest accuracy for the female samples was achieved on the two words titles, while the lowest accuracy for the male samples was achieved on the three words titles.

In contrast with previous results (Figure 5), it seems that the Commands model (Figure 6) does not show any kind of bias, in terms of accuracy, on the one-word movie titles and on the longest movie titles. Looking at the two and three words titles, it seems the model shows a bias towards the female voices, in terms of accuracy. However, the WER shows that the model did not perform that bad on the male audios. The WER on the two words titles is slightly worse on the male samples than on the female samples. However, on the three words

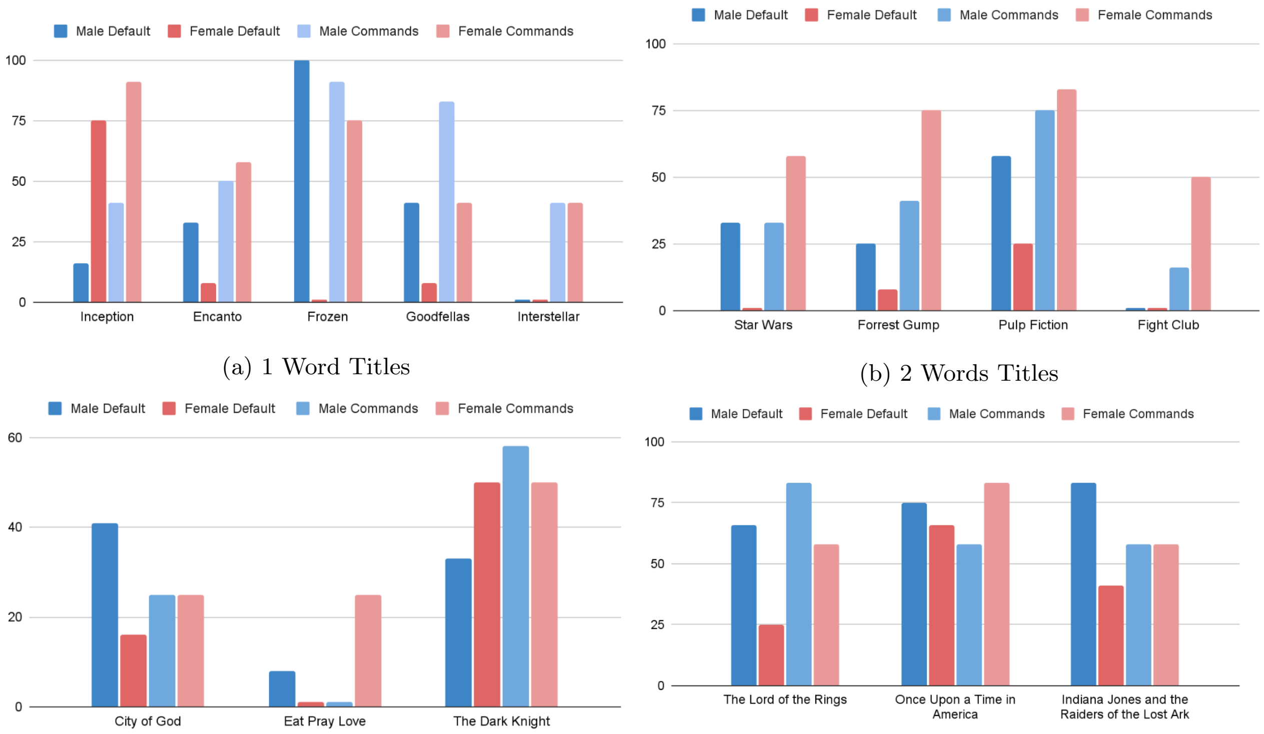
(a) Accuracy

Words	Female	Male	Discrimination
1	61%	61%	0%
2	66%	41%	-23%
3	33%	27%	-10%
3+	66%	66%	0%

(b) Word Error Rate

Words	Female	Male	Discrimination
2	0.39	0.45	-7%
3	0.61	0.52	7%
3+	0.26	0.27	1%

Table 6: Results on the Movie Titles dataset using the Commands Model



(a) 1 Word Titles

(b) 2 Words Titles

(c) 3 Words Titles

(d) 3+ Words Titles

Figure 4: Accuracy on the Movies Title dataset

titles, the WER is slightly better on the male samples. On average, the WER discrimination is almost zero on the movie titles dataset. When calculating the accuracy, a positive result is considered only when all the words from the movie title are matched. Thus, having a higher accuracy for the female audios, means that indeed it fully matched more titles than on the male samples, however, looking at the WER it seems that the results on the male samples only lacked very few words on a sample to be considered a perfect match.

Comparing the performance of the two models on the male voices, the Default model performed slightly better. In contrast, on the female samples, the Commands model had a performance twice as big as the Default model. Looking at the WER, the performance on the male samples is the same between the two models, but on the female samples, the Default model reported a WER almost twice as big as the Commands model.

In terms of mistakes, the Default model had very few deletions, but a lot of substitutions and insertions. In contrast, the Commands model rather suffered from deletions and very few substitutions. Mistakes across repetitions of experiments were fairly consistent. For some of the speakers, the Commands model ignored all audio samples on a certain movie title. One could blame the quality of the recording, but given that using the same sample with the Default model had a few matches, I do not think that is the case. Mistakes between the two genders were quite inconsistent, meaning that the model did not make the same substitutions. For example for ”Interstellar”, substitutions on the female voices are as follows: ”Stella”, ”Estella”, ”Isabella”, and substitutions on the male voices are: ”Stellar”, ”Eller”, ”Internal”. It seems that the substitutions on the female voices lack the ”r” letter. Also, the mistakes are rarely consistent within the female speakers group and the male speakers group, but when a speaker repeats a movie title, sometimes it makes the same substitution or deletion on all repetitions.

Discussion and Conclusion

The Default model of the Twilio ASR system proved to be biased toward male voices on both the Google Commands dataset and on the Movie Titles dataset. In contrast, the Commands model offered by Twilio showed no bias towards either of the genders on the Movie Titles dataset.

One could assess the difference in performance to the quality of the audio files. The audio files provided on the Google Commands dataset for the female voices were indeed less qualitative than the one with the male voices. In the female samples, there was a lot of background noise and most of the speakers talked with accents. In some of the samples, the female voices talked very fast and less clear. However, the Default model showed bias towards the male voices on two different datasets. Moreover, the Commands model had no problem detecting the voices of the females from the Movie title dataset, indicating that the quality of the samples was not a problem. In addition, it is to be noted that some of the speakers recorded the samples in the same surroundings with the same type of background noise, making the quality of the samples pretty similar between the two gender groups.

Most of the researchers assess this bias due to the imbalance in the training dataset between female and male samples [7] [8]. Having a training dataset composed mainly of male samples and very few female samples can lead to gender bias [4]. That is because the model learns to minimize the loss based on the majority group. The model could learn a pattern that distinguishes the male voices from the female voices and thus without specifying the gender, it can learn to separate the majority male group from the female group, favouring the majority male group when training. Male and female voices differ in terms of pitch, pronunciation, tone, intensity etc., thus it is easy for a model to learn to separate the two groups. Although the quality of the samples is not the problem in the bias of the Default model, one should try to gather samples that are equally qualitative for both female voices and male voices. Having a model learn from less qualitative data for a certain group could lead to bias.

No information was provided regarding the training dataset of the Default model, thus the reason for the presence of bias is unknown. Further research could focus on experimenting with more samples for both the female and male voices, but considering the limitations of the trial account, this was not possible at the moment. An interesting future work could be to investigate the impact of the voice's features. Thus, experimenting with high pitch, low pitch etc. samples for both female and male voices to assess which one has the biggest impact on the presence of bias.

In conclusion, when creating an ASR system, one should test not only its overall performance but also the performance on gendered groups to try to assess if the model presents a significant and consistent bias. The Cinema Ticket Reservation System created using the ASR system provided by Twilio presented a slight bias towards the male group using the Default speech model. As mentioned before, it is important to test for such biases, especially in commercial speech recognition systems that are meant to ease the life of the user, as all users should benefit from the same user experience. To avoid having a biased ASR system, one should create a training dataset equal in samples and quality between the female and male samples. It is to be noted that when testing for bias, one should test on an equal number of samples and equally qualitative samples.

References

[1] *About VoiceXML and Voice Browsers.* `https://www.dialogic.com/webhelp/CSP1010/VXML1.1CI/WebHelp/intro%20-%20About%20VoiceXML%20and%20Voice%20Browsers.htm`. Accessed: 2022-06-06.

[2] *Amazon scraps secret AI recruiting tool that showed bias against women.* `https://www.reuters.com/article/idUSKCN1MK08G`. Accessed: 2022-06-06.

[3] Marcely Zanon Boito et al. "A Study of Gender Impact in Self-supervised Models for Speech-to-Text Systems". In: *arXiv preprint arXiv:2204.01397* (2022).

[4] Alexandra Chouldechova and Aaron Roth. "A snapshot of the frontiers of fairness in machine learning". In: *Communications of the ACM* 63.5 (2020), pp. 82–89.

[5] Ross Corkrey and Lynne Parkinson. "Interactive voice response: review of studies 1989–2000". In: *Behavior Research Methods, Instruments, & Computers* 34.3 (2002), pp. 342–353.

[6] Marta R Costa-jussà, Christine Basta, and Gerard I Gállego. "Evaluating gender bias in speech translation". In: *arXiv preprint arXiv:2010.14465* (2020).

[7] Mahault Garnerin, Solange Rossato, and Laurent Besacier. "Gender representation in French broadcast corpora and its impact on ASR performance". In: *Proceedings of the 1st International Workshop on AI for Smart TV Content Production, Access and Delivery.* 2019, pp. 3–9.

[8] Mahault Garnerin, Solange Rossato, and Laurent Besacier. “Investigating the Impact of Gender Representation in ASR Training Data: a Case Study on Librispeech”. In: *3rd Workshop on Gender Bias in Natural Language Processing.* Association for Computational Linguistics. 2021, pp. 86–92.

[9] *Google’s Speech Commands Dataset.* `https://pyroomacoustics.readthedocs.io/en/pypi-release/pyroomacoustics.datasets.google_speech_commands.html`. Accessed: 2022-06-06.

[10] Allison Koenecke et al. “Racial disparities in automated speech recognition”. In: *Proceedings of the National Academy of Sciences* 117.14 (2020), pp. 7684–7689.

[11] Graeme McMillan. “It’s not you, it’s it: voice recognition doesn’t recognize women”. In: *Retrieved on February* 4 (2011), p. 2017.

[12] *Microsoft’s AI chatbot, gets a crash course in racism from Twitter.* `https://www.theguardian.com/technology/2016/mar/24/tay-microsofts-ai-chatbot-gets-a-crash-course-in-racism-from-twitter`. Accessed: 2022-06-06.

[13] James A Rodger and Parag C Pendharkar. “A field study of the impact of gender and user’s technical experience on the performance of voice-activated medical tracking application”. In: *International Journal of Human-Computer Studies* 60.5-6 (2004), pp. 529–544.

[14] José Rouillard. “Web services and speech-based applications around VoiceXML.” In: *J. Networks* 2.1 (2007), pp. 27–35.

[15] Arjun Sahani et al. “Speech Recognition for Cinema Ticket Booking System”. In: *International Journal of Engineering Research in Computer Science and Engineering* 5.3 (2018).

[16] Rachael Tatman. “Gender and dialect bias in YouTube’s automatic captions”. In: *Proceedings of the first ACL workshop on ethics in natural language processing.* 2017, pp. 53–59.

[17] *TwiML for Programmable Voice.* `https://www.twilio.com/docs/voice/twiml`. Accessed: 2022-06-06.

[18] Alexander Waibel and Kai-Fu Lee. *Readings in speech recognition.* Morgan Kaufmann, 1990.