The Effect of Noise Overlay on the Performance of ASR Models

Mikhail Bulygin

July 2020

1 Introduction

1.1 Motivation

The field of Automatic Speech Recognition came a long way 18th century with Wolfgang von Kempelen's Acoustic Mechanical Speech Machine [4] and evolving to today's digital assistance and smart home technologies [6]. With the emergence of new advancements in the field came new concerns. People started to worry that these new systems can cause harm, threatening one of the basic human rights of privacy [14]. The public got even more agitated after Edward Snowden's case that revealed that the US government has instituted the system of mass surveillance, and was tracking and collecting private information of its citizens [15].

This sense of distrust for technology created demand for a safer environment. New regulations, concerning privacy and ethics, were put in place. GDPR law was adopted in 2016 and is governing the use of private information in the EU. US law system is more complex and diverse, but in various states, this issue is being addressed [1].

Yet all these new regulations were not able to make the public feel secure and new adversarial products, like Alias [7] were created. Project Alias as described by its creator Bjørn Karmann is "a teachable

"parasite" that is designed to give users more control over their smart assistants". Alias is placed on top of the microphone of the smart assistant, and once it is turned on it is sending white noise that 'silences the ears' of the speech recognizer. It reacts to the keyword that the user sets manually, and once this word is pronounced it turns off the white noise and allows the smart assistant to listen to the user.

Another concern that the public expressed was the possible threat of the home assistant being hacked. The owner might never notice that his system was broken in since the ASR systems can be redesigned to recognize the input from the ultrasonic range [17], which is inaudible for humans.

These threats and the emergence of adversarial technology brought my attention to speech recognition falsification. In this thesis research, we are going to look at how much do various noises corrupt the initial audio and how do initial input affects this process.

1.2 Background

Even considering the recent breakthrough of the ASR models, they are still are highly dependable on the quality of the input source [8]. In this study, we will try to exploit that weak spot of speech recognition models to experiment on the falsification of ASR systems.

Due to the recent realization of the possible threat of ASR systems and also to some degree reverse intuition of such research, there is very little research on that topic known to us. However, we can take inspiration from the research conducted on the noise-resistant ASR models. The standard training dataset for training noise-robust models is Aurora dataset [13]. Aurora consists of the 7,138 utterances from the Wall Street Journal (WSJ0) Corpus. Each of these utterances is transformed with the addition of noise. Aurora uses six real-life noise conditions (street traffic noise, train station, restaurants, etc) added to clean recordings through two channels: overlaying noise and recording with a separate microphone.

As was already mentioned, the Aurora dataset contains recordings of utterances with real-life noise overlay. That makes sense since this

dataset was designed to help advancement in the noise-resistant ASR models. These models need to perform well in ordinary noisy environments, which can occur in real life. However, for our task, we can investigate further and also experiment with the artificial noises (e.g. white noise that was used in the Alias system).

The two types of noise overlay were investigated in the study of adversarial examples for ASR [16]. In this paper, a group of researchers from Ruhr University advocates against the use of over-the-air attack (separate microphones). In our research, we planned to experiment with two types of overlay, but due to the COVID-19 pandemic, we couldn't experiment with separate microphones set-up. Nevertheless, we feel that the research does not significantly lose in value, as originally the idea for the study was inspired by the Alias model, which feeds noise directly into the ASR system.

1.3 Research question

In the present study, we would like to investigate the effect of noise overlay on the ASR systems. More specifically, we are interested in which noise can corrupt the original input the most. This study includes experiments with artificial noises (white noise, pink noise, etc.) and real-life noise (street sounds, construction noise, etc.).

Moreover, we will look into the effects of noise overlay on different types of input data. Our training dataset includes information on the speaker's gender that will be incorporated in the analysis part of this research.

Overall, this work poses two research questions:

- What kind of noise used in the experiment can compromise the performance of the ASR system the most?
- Does the gender of the speaker correlate with the effect of noise on the performance of the ASR system?

2 Methods

2.1 Speech Recognition Model

Our main goal in this research is to measure the effect of various noises on the performance of ASR. Given that, we decided that the best solution would be to use already existing and popular ASR systems for testing of noise effects. For our task, we selected ASR models from the SpeechRecognition python library [18], because it contains many popular ASR distributions and it is easy to use. In total, SpeechRecognition library supports 8 speech recognition APIs:

- CMU Sphinx
- Google Speech Recognition
- Google Cloud Speech API
- Wit.ai
- Microsoft Bing Voice Recognition
- Houndify API
- IBM Speech to Text
- Snowboy Hotword Detection

In our research, we have used only three distributions: CMU Sphinx, Google Speech Recognition, and Wit.ai. We chose these APIs because they are distributed free of charge and do not require any type of subscription, unlike the rest of the models.

CMU Sphinx is an API developed at Carnegie Mellon University. It uses an LVCSR method and can be used in an offline mode [5]. It is constantly updated and the source code can be found on the projects GitHub page.[1]

[1] https://github.com/cmusphinx

Google Speech API was trained using attention-based architecture for sequence to sequence speech recognition [3]. This model uses acoustic, pronunciation, and language models for its input into the neural network. The model shows great performance and also had the fastest recognition time in our experiment.

Wit.ai is a company that is part of the Facebook corporation. Their main goal is to provide developers with convenient tools for developing language apps. At the moment the model is not openly distributed, and we don't know the methods used in its training [9]. Nevertheless, this API shows good performance, so we decided to include it in the experiment.

2.2 Metrics

To measure the influence of the noise, first, we have evaluated the performance of speech recognition models on the clean speech, and then the same operation was conducted on the noisy data.

The results for the models are measured using several metrics. First, we use the Word Error Rate (WER), which is a common assessment method for ASR models [11]. WER looks at the number of substitutions, insertions, and deletions in the ASR prediction, and compares it with the ground truth. To get more insight into the effect of the noise we are also applying the Word Information Lost (WIL) metric, which is based on mutual information between the prediction and ground truth [11]. Both metrics have a range from 0 to $+\infty$, however, usually, the score falls between 0 and 1, with 0 being a perfect score.

Also, we decided to include the Bilingual Evaluation Understudy (BLEU) metric. That score is commonly used for machine translation, but in our opinion, it can give us valuable information for the task of ASR. The BLEU metric has a range between 0 and 1 with 1 being a perfect match and 0 – the worst score possible. To make the metric homogeneous we flipped the result for the BLEU score.

To compute the WER and WIL results for the speech recognition we have used JiWER library [2] that provides the tools for the evalua-

tion of ASR models. For the BLEU metric we have used NLTK python package [10].

2.3 Experiment outline

To find the answers to our research questions we have conducted a set of experiments. First, prepared the audio data. The clean speech utterances are taken from the LibriSpeech dataset [12] and to create the noisy data we overlay this audio with noise samples. Next, we run the resulting audio through the ASR models mentioned above. We compute the scores for all the data and compare the results. The type of noise that causes the most significant drop in recognition quality is the most efficient noise for the falsification of the ASR model's performance.

We can summarize the experiment outline with the following steps:

1. Load the clean utterances
2. Overlay clean utterances with noise
3. Run the ASR models on the resulting data
4. Compute the scores for all the utterances
5. Compare the results of noisy data and clean data

3 Set-up

3.1 Speech and Noise Data

To test noise effects we needed the annotated speech data to which we would apply noise. Our choice fell on the LibriSpeech dataset [12], which contains 1000 hours of read English speech. The data for the corpus is taken from the audiobooks of the LibriVox project.

Due to huge processing requirements for speech recognition, we decided to take only a small subset of that corpus. Our data consists of

1000 examples from the 'clean' subcorpus of the LibriSpeech dataset, aligned with text transcriptions of the speech. Also, the annotation contains information on the sex of the speaker, which will be used in the analysis of this research.

There are 16 different speakers, 6 of whom are males and 10 are females. All of them are native speakers of English. The samples in our subcorpus range from 4s to 20s long.

We chose the LibriSpeech corpus for our experiment, because it contains very clear, well-pronounced utterances. This kind of data is perfect for the baseline results because there is no disturbance that can throw the ASR model off.

Noise data was collected by us from the audiocheck[2] website, which distributes various audio data free of charge. For our experiment, we have used both artificial and real-life noises. Artificial noises include white noise, pink noise, brown noise, blue noise, and violet noise. The real-life noises in our study are street sounds, conversation noise, dog barking noise, and roadwork noise. Also, we decided to experiment with the input data itself and overlay the speech with the input audio broken down and shuffled randomly. The example samples of the noises can be found on the project's GitHub page[3].

The data used for testing of ASR models was taken from the 'clean' subcorpus of LibriSpeech dataset. For our experiment we have used 1000 randomly sampled utterances,

3.2 Adding Noise

Noise samples were taken from the internet and were looped onto the speech input. For our work, we decided to set the single value for the Signal to Noise Ratio (SNR) of 10. We chose to use one value for SNR to make sure the experiment has no bias. The value of 10 is a common setting for the SNR and is used in the Aurora dataset and other research on the topic [13] [16].

[2] https://www.audiocheck.net/

[3] https://github.com/bulyginmv/ASR_Noise

To illustrate the effect of noise overlay we include the side by side comparison of the Mel-frequency cepstral coefficients (MFCC) of the same sample (Figure 1). The left side contains clean utterance and the right side includes noise overlay.

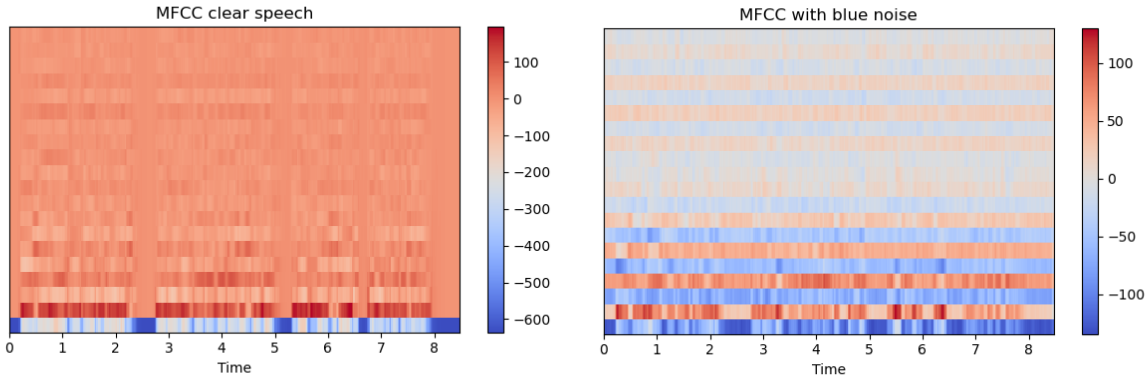

Figure 1: A MFCC comparison of clean and noisy data. Part 1

As we can see the difference between the MFCCs is significant and can be visually spotted when blue noise is applied.

The difference is not as drastic, but still quite noticeable when we overlay the audio with roadwork noise (Figure 2)

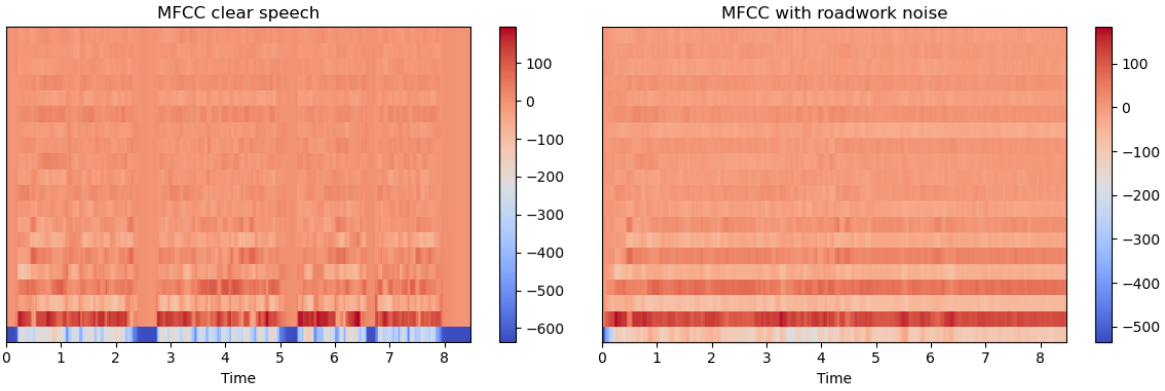

Figure 2: A MFCC comparison of clean and noisy data. Part 2

4 Experiment

To conduct the experiment we wrote a python script. The program takes 1000 utterances from the LibriSpeech corpus. For each speech fragment, the algorithm performs a loop applying 10 noises to it. Each noisy audio fragment, as well as clean speech, is run through the ASR model. The prediction is compared with ground truth annotation of the utterance and the WER, WIL, and BLEU scores for each audio is computed and stored in the Pandas table. That script is repeated for all three ASR models. We decided to not apply any denoising techniques to keep the experiment unbiased for the models.

The WER and WIL metrics are calculated using standard parameters. The BLEU score is calculated with 0.7 weight for the unigrams, 0.25 weight for the bigrams, 0.04 weight for the trigrams, and 0.01 for the quadrigrams.

The Pandas table consists of 35 columns. We present the directory for each file, the 3 scores for each type of audio, and the sex of the speaker. The CSV files for all the models tested in the experiment can be found on the GitHub page of the project.

The models' execution time varied. For 1000 utterances Google Speech API was the fastest, finishing recognition in about 7 hours. The execution of the Sphinx model took much longer. The same task took around 22 hours to complete. And Wit.ai model finished in approximately 12 hours.

5 Analysis & Results

After all three models finished recognizing the speech data, we calculated the mean value for every score for each type of noise. Next, we calculated the difference between the score for the clean audio and the noisy score. The results are presented in Table 1. The higher the value in the table the more the noise corrupted the recognition of the audio.

	Google API			Sphinx API			Wit.ai API		
Noise	WER	WIL	BLEU	WER	WIL	BLEU	WER	WIL	BLEU
White Noise	**0.357**	**0.376**	**0.393**	**0.675**	**0.599**	**0.659**	0.472	**0.429**	0.474
Pink Noise	0.295	0.315	0.327	0.66	0.588	0.649	**0.48**	0.426	**0.479**
Brown Noise	0.014	0.017	0.02	0.087	0.107	0.099	0.109	0.117	0.142
Blue Noise	0.252	0.282	0.287	0.633	0.586	0.646	0.356	0.353	0.379
Violet Noise	0.17	0.2	0.192	0.571	0.548	0.601	0.271	0.288	0.304
Dog Bark	0.115	0.125	0.133	0.388	0.388	0.38	0.325	0.324	0.356
Speech	0.38	0.373	0.426	0.574	0.574	0.636	**0.546**	**0.469**	0.521
Streets	0.17	0.187	0.19	0.508	0.497	0.562	0.411	0.373	0.435
Roadworks	**0.393**	**0.396**	**0.439**	**0.651**	**0.585**	**0.647**	0.528	0.461	**0.522**
MixTalk	0.304	0.292	0.348	0.558	0.462	0.479	0.378	0.372	0.411

Table 1: Difference between the mean score of clean speech and noisy speech

We highlighted the highest scores for each metric for every model. The noise that compromised the data the most was the sound of the construction work. This noise got the highest result for 7 out of 9 experiments.

For the most part, real-life noises outperform the artificial ones. In terms of artificial noises, white noise shows the best result. It also topped the list for the artificial noises in 7 out of 9 experiments.

Other noises that scored high in our experiments were the speech of other people and the pink noise. Both of them showed the highest results for Wit.ai API. Using the input itself as the noise constructer did not show to be as effective as other noises, with MixTalk showing average performance in comparison with top noises.

It is interesting to note that brown noise had very little effect on the speech data. The corruption level is very low, especially in comparison with other noises.

Another thing worth mentioning is that the results seem to consistent across all three models. The same noises show the highest scores, and the noises with low scores perform accordingly for each API. That allows us to believe that the results of our experiment can be expended for other ASR models too.

Overall, in our experiment, Google Speech API showed to be the

most noise resistant model. The model most dependable on the quality of noise is the CMU Sphinx model. In Figure 3 we present the box plot for the Google Speech API. Similar plots for other models can be found on the project's GitHub page.

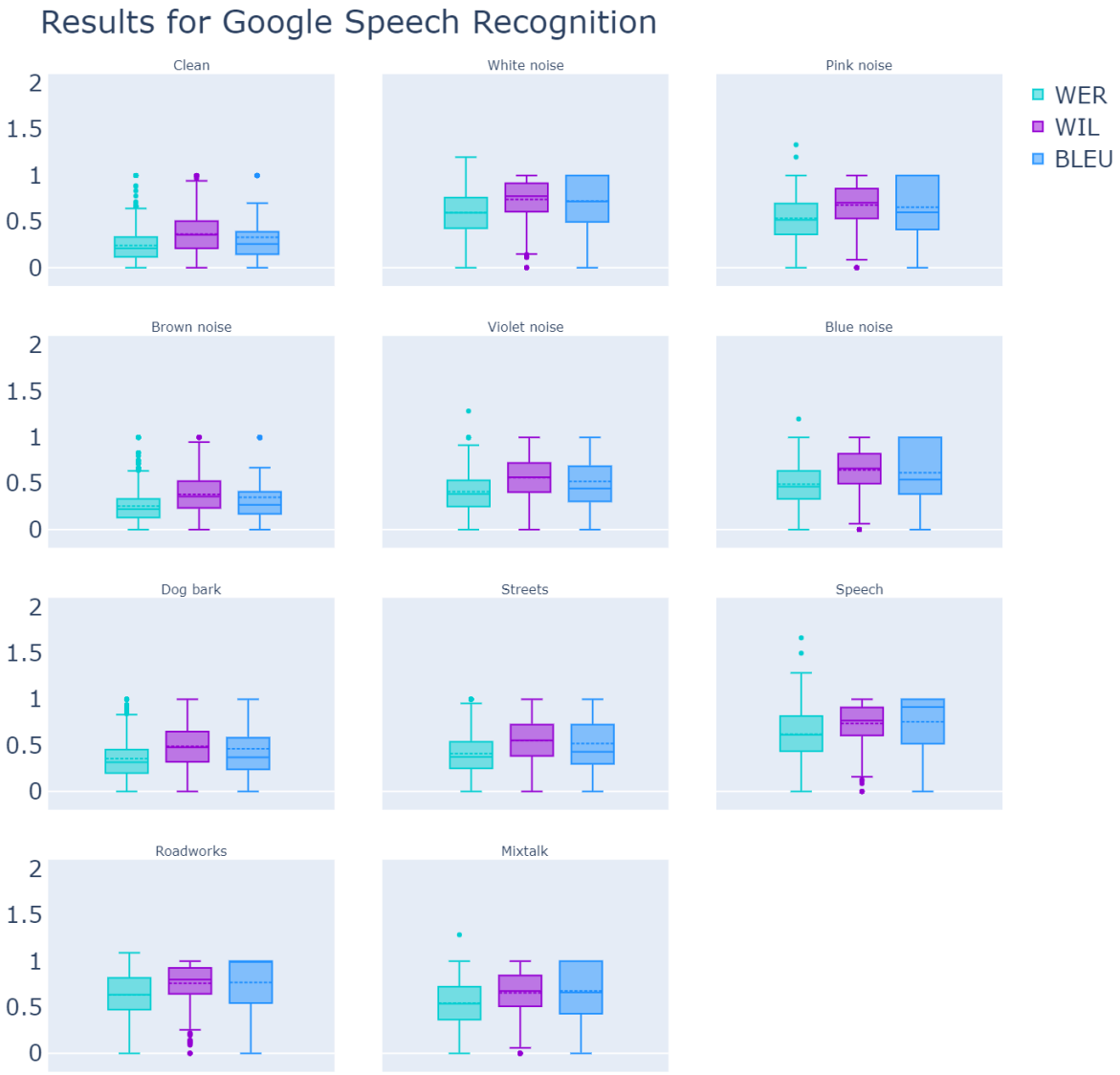

Figure 3: A box plot for the Google Speech API

To answer our second research question, we looked at way the noise overlay correlated with the gender of the speaker. We devided all samples in two groups one read by males and the second read by females. After that we calculated the mean score for each type of audio. For example, in Table 2 the results for Wit.ai Speech API are presented.

	WER		WIL		BLEU	
Noise	Male	Female	Male	Female	Male	Female
Clean Audio	0.328	0.326	0.456	0.451	0.423	0.432
White Noise	0.837	0.776	0.917	0.861	0.942	0.877
Pink Noise	0.849	0.78	0.918	0.854	0.949	0.881
Brown Noise	0.45	0.427	0.591	0.557	0.598	0.554
Blue Noise	0.61	0.59	0.759	0.73	0.745	0.725
Violet Noise	0.704	0.67	0.836	0.788	0.842	0.787
Dog Bark	0.641	0.658	0.783	0.774	0.783	0.785
Speech	0.783	0.709	0.872	0.797	0.916	0.83
Streets	0.911	0.849	0.953	0.902	0.982	0.929
Roadworks	0.884	0.836	0.94	0.898	0.98	0.932
MixTalk	0.683	0.717	0.82	0.828	0.838	0.84

Table 2: Comparison of male and female speakers' scores for Wit.ai Speech API

Similar scores were calculated for the other two models. To conclude if there is a bias for one of the genders, we calculated the p-value between two sets. The speaker's gender showed to not correlate with the effect of noise overlay. The p-value for Google Speech API was 0.98. A slightly lower, but still well over the threshold result of 0.7 was calculated for the CMU Sphinx model. The Wit.ai Speech API got the lowest value of 0.39. Still, we can conclude that the effect of noise on the corruption of the audio does not depend on the gender of the speaker.

6 Discussion

The experiments conducted in this study show that noise overlay can cause massive problems for automatic speech recognition systems. Both

artificial and real-life noises are capable of corrupting the original audio signal. Though, not all noises produce the same result. The noise that effected the clean audio signal the most was the sample of the road-works noise. In the domain of artificial noises, the best performance was shown by the white noise.

Also, we proved that the effects of the noise on the ASR systems listed in our experiment have no bias towards the gender of the speaker in the original audio sample. We prove that there is no correlation by conducting a p-value test.

Possible future development of this research is the study of psychoacoustic masking [16]. It would be interesting to see if it is possible to hide the noise from the human's hearing range and still maintain the corruption effect. That research would go in line with the projects like Alias, that are designed to destroy the input for the ASR system, and at the same time not annoying the user with the sidenoise.

There are a couple of limitations to our project. First of all, due to the short amount of time and low processing power we had did not use an extensive utterance dataset for the experiment. It is possible that given more audio samples we can see a change in the results. Also, due to the limited resources we only experimented with overlaying of the noise and did not try the over-the-air attack approach. Using different set-up might bring out new insights for our study.

In this research, we tested only three ASR models. Even in this small sample, we see some discrepancies between different systems. It seems that noise-resistance is highly dependable on the design of the model. Although in our experiment the results seem to exhibit the same tendencies across all the models, the results might change for models that use different approaches to speech recognition.

7 Conclusion

This study presented an experiment on the effect of noise overlay for the performance of the ASR systems. We used utterances from the LibriSpeech corpus and added 10 noise samples to the clean speech

fragment. The resulted audio used as an input for three ASR models: Google Speech API, CMU Sphinx API, and Wit.ai Speech API. Using the predictions from the models we calculated the scores over three metrics: Word Error Rate, Word Information Lost, and Bilingual Evaluation Understudy. The results were analyzed and compared with the scores for the clean speech samples. Our experiment demonstrated that the speech recognition APIs listed in our research are vulnerable to the addition of noise to the original audio. The noise of roadworks showed to compromise the performance of ASR models the most. Among the artificial noises, white noise expressed the most corrupting effect of the original audio. The study of the speaker's gender bias towards the effect of noise proved to be negative. At the end of the thesis, we discuss possible limitations of the study and propose future developments for the research of this topic.

References

[1] California Consumer Privacy Act: 2019 Final Amendments Signed. https://www.gibsondunn.com/california-consumer-privacy-act-2019-final-amendments-signed/. Accessed: 2019-10-16.

[2] JiWER: Similarity measures for automatic speech recognition evaluation, 2020. Available from https://github.com/jitsi/jiwer.

[3] Chung-Cheng Chiu, Tara N. Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J. Weiss, Kanishka Rao, Katya Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani. State-of-the-art Speech Recognition With Sequence-to-Sequence Models. *CoRR*, 2017.

[4] Homer Dudley and T. H. Tarnoczy. The Speaking Machine of Wolfgang von Kempelen. *The Journal of the Acoustical Society of America*, 22(2):151–166, 1950.

[5] D. Huggins-Daines, M. Kumar, A. Chan, A. W. Black, M. Ravishankar, and A. I. Rudnicky. Pocketsphinx: A Free, Real-Time Continuous Speech Recognition System for Hand-Held Devices. In *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, 2006.

[6] Yousra Javed, Shashank Sethi, and Akshay Jadoun. Alexa's Voice Recording Behavior: A Survey of User Understanding and Awareness. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, New York, NY, USA, 2019. Association for Computing Machinery.

[7] Bjørn Karmann. Project Alias. http://bjoernkarmann.dk/project_alias. Accessed: 2018.

[8] Davis Liang, Zhiheng Huang, and Zachary C. Lipton. Learning Noise-Invariant Representations for Robust Speech Recognition, 2018.

[9] Jason Liao. Hello speech, we love you too. https://medium.com/wit-ai/hello-speech-we-love-you-too-5851d0b34f9f. Accessed: 2019.

[10] Edward Loper and Steven Bird. NLTK: The Natural Language Toolkit. In *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*, page 63–70, USA, 2002. Association for Computational Linguistics.

[11] Andrew C. Morris, Viktoria Maier, and Phil D. Green. From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition. In *INTERSPEECH*, 2004.

[12] 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.

[13] N. Parihar and J. Picone. DSR Front End LVCSR Evaluation - AU/384/02, Aurora Working Group, European Telecommunications Standards Institute, 2002.

[14] Oleksandr Pastukhov and Els Kindt. Voice Recognition: Risks To Our Privacy. https://www.forbes.com/sites/realspin/2016/1006/voice-recognition-every-single-day-every-word-you-say/#6793e919786d. Accessed: 2016-10-06.

[15] Dinah PoKempner. US Government Mass Surveillance Isn't 'Secret'. https://www.hrw.org/news/2019/09/18/us-government-mass-surveillance-isnt-secret. Accessed: 2019-09-18.

[16] Lea Schönherr, Thorsten Eisenhofer, Steffen Zeiler, Thorsten Holz, and Dorothea Kolossa. Imperio: Robust Over-the-Air Adversarial Examples for Automatic Speech Recognition Systems, 2019.

[17] Chen Yan, Xiaoyu Ji, Tianchen Zhang, Taimin Zhang, Wenyuan Xu, and Guoming Zhang. Dolphin Attack: Inaudible Voice Commands, Oct 2017.

[18] A. Zhang. Speech Recognition (Version 3.8), 2017. Available from https://github.com/Uberi/speech_recognition#readme.