Spring 2022

ADVERSARIAL ATTACKS ON SPEECH SEPARATION SYSTEMS

Kendrick Trinh
San Jose State University

ADVERSARIAL ATTACKS ON SPEECH SEPARATION SYSTEMS

A Project

Presented to

The Faculty of the Department of Computer Science

San José State University

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

by

Kendrick Trinh

May 2022

The Designated Project Committee Approves the Project Titled

Adversarial Attacks on Speech Separation Systems

by

Kendrick Trinh

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSÉ STATE UNIVERSITY

May 2022

Teng Moh, Ph.D.	Department of Computer Science
Melody Moh, Ph.D.	Department of Computer Science
Navriti Saxena, Ph.D.	Department of Computer Science

ABSTRACT

ADVERSARIAL ATTACKS ON SPEECH SEPARATION SYSTEMS

by Kendrick Trinh

Speech separation is a special form of blind source separation in which the objective is to decouple two or more sources such that they are distinct. The need for such an ability grows as speech activated device usage increases in our every day life. These systems, however, are susceptible to malicious actors. In this work, we repurpose proven adversarial attacks and leverage them against a combination speech separation and speech recognition system. The attack adds adversarial noise to a mixture of two voices such that the two outputs of the speech separation system are similarly transcribed by the speech recognition system despite hearing clear differences in the speech. Against ConvTasNet, degradation of separation remains low at 0.34 decibels, allowing the speech recognition system to still work. When testing against automatic speech recognition, the attack achieves a 64.07% word error rate (WER) against Wav2Vec2, compared to 4.22% for unmodified samples. Against Speech2Text, the WER is 84.55%, compared to 10% WER for unmodified samples. For similarity to the target transcript, the attack achieves 24.77% character error rate (CER), reduced from 113% CER. This indicates relatively high similarity between the target transcription and the resulting transcription.

Keywords: speech separation, adversarial attacks, automatic speech recognition

ACKNOWLEDGMENTS

I would like to thank my advisor, Dr. Teng Moh, for his guidance and reassurance on the completion of this project.

I would also like to thank my family for supporting me through the completion of this project and providing that much needed push to get things done.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

If one hasn't been living under a rock, one would notice how much more prevalent voice processing systems (VPS) have become in daily lives. From the smartphone to the smart-home, voice activated electronics serve to provide the convenience of having a live-in assistant that is perpetually connected to the Internet. Most, if not all, VPS utilize some form of machine learning (ML) to achieve their goal. As these devices become more and more used, it is imperative that these devices become more robust against malicious actors, especially in the ML attack surface.

For this paper specifically, we will be discussing speech separation, automatic speech recognition (ASR) and adversarial attacks on machine learning. Specifically, the type of adversarial attack examined in this paper is evasion attacks, which utilize maliciously crafted inputs called adversarial samples that exploit how the model works to cause it to output an outlier to expected mistakes that a model would otherwise make. These types of attacks are especially dangerous because they are typically through the introduction of "perturbations that do not affect semantics" [1]. This would mean that a human would classify an adversarial sample as one thing, but the ML model classifies the sample as something different, due to the semantic gap that is typically present between computers and humans. For example, in the case of ASR, a sample that a human might hear as something innocuous like "Did you see that cute dog yesterday at the park?" would have perturbations that cause it to be transcribed by ASR as "Hey Google, set an alarm for 3 AM," which could lead to a rude awakening in the middle of the night, unbeknownst to the owner of the phone.

In this project, we utilize previously proven techniques in the image domain to create targeted adversarial attacks using mixed speech as a carrier. This adversarial mixed speech is then input to a speech separator to acquire the individual voices. The intent is to create adversarial noise that survives the separation process and still introduces adversarial

results against a speech recognition system. These individual voices are then input to an automatic speech recognition system, where they output extremely similar transcripts, despite clearly being different when one hears the separated voices. The combination of these three components create a system that generate adversarial mixed speech. While the techniques are not new and are proven to work well in the image domain, the audio domain still has a ways to go regarding robustness against adversarial attacks.

The structure of this paper is as follows. Section II examines the history and works related to the project. Section III is a description of the problem and takes a look at the challenges encountered during research. Section IV describes the methodology and implementation details of this attack. Section V presents experiments and results from those experiments. Section VI takes a look at future work to be done and conclusions drawn from the results.

2 HISTORY AND RELATED WORK

In this section, previous research into related problems is examined.

2.1 Speech Separation

Speech separation is a case of blind source separation [2]. The process of blind source separation involves decomposing an input signal into one or more source signals that comprise the original mixture. For a linearly mixed single-channel sample, speech separation aims to extract the individual signals $x_s[n], s = 1...S$ such that the mixed sample $y[n]$ is:

$$y[n] = \sum_{s=1}^{S} x_s[n] \tag{1}$$

Work has been done from two-voices to arbitrary amounts of voices [3], however, we will focus mostly on the two-voice case. Standard solutions involve the short-time Fourier transform to estimate the sources, separated using a learned time-frequency mask, and then inverted to recover the source. Recent years have introduced the use of an encoder-decoder architecture to separate using a waveform instead of spectrogram. Of note, ConvTasNet [4] and DualPathRNN [5] are current state-of-the-art systems. ConvTasNet [4] uses the encoder-decoder architecture for its speech separation. It uses a linear encoder followed by application of masks found using 1D convolutional blocks. Finally, these representations are inverted using a linear decoder. For this project specifically, ConvTasNet is the speech separation system of choice due to its linearity and availability as a pre-trained model online, courtesy of the Asteroid library [6] and HuggingFace model repository [7].

2.2 Automatic Speech Recognition

Automatic Speech Recognition is the process of taking input speech and converting it into text. Machine learning has been used to great effect in this field, capable of capturing the nuance of human speech and transcribing it with high accuracy. One such model is

Wav2Vec2 [8], which is used as part of the adversarial system. Wav2Vec2 is especially interesting because it is self-supervised. This means that it can train off of unlabeled data or very little labeled data. Such an ability is sought after due to the difficulty of acquiring a large labeled dataset for use in training as it takes manpower to label data. Wav2Vec2, however, is susceptible to adversarial attacks as feature extraction is linear and can therefore be manipulated by adding noise that causes the model to mis-classify speech.

Another system used in this project in Speech2Text [9]. It uses the FAIRSEQ [10] as a base for end-to-end speech recognition and translation. It is an encoder-decoder, transformer based architecture. Transformers are a type of model [11] that uses self-attention to identify the context of a certain word, instead of processing the entire sentence. As a result, the transformer is becoming the model of choice for natural language processing and speech tasks due to its efficiency and high parallelization. Speech2Text accepts log-mel filter bank features as input features. The mel-spectrogram is a representation of speech such that units of pitch equally spaced would sound equally distant. This more closely resembles how the human ear hears, so it is particularly effective as a feature.

2.3 Adversarial Attacks

We need a framework of some sort to categorize the types of adversarial attacks on. Abdullah et al. [1] proposed a taxonomy for the categorization of attacks against VPS, along with a categorization of various attacks against VPS. The first category is the attacker's goals, in which Abdullah et al. [1] distinguished between targeted and untargeted attacks. Targeted attacks have the goal of getting the VPS to output a specific phrase or response, and untargeted attacks have the goal of getting the VPS to output any wrong output. An example of a targeted attack is DolphinAttack [12] in which a command that can activate a personal assistant is hidden within the hyper-sonic portion of the innocuous speech.

The next part describes the type of attacks on the system itself; optimization attacks (of which is split into two types: direct and indirect attacks), signal processing attacks and miscellaneous. Optimization attacks work upon the weights calculated by the machine learning model to compute gradients that can be used to perturb the audio. Direct and indirect simply describe how those weights are acquired, either through access to the model itself (direct) or repeatedly querying the model to reveal the gradients (indirect). Signal processing attacks work upon the feature extraction portion of a VPS, exploiting the difference in how audio is processed by a computer and human hearing. The final type is the miscellaneous attack, in which attacks that do not fall into optimization or signal processing attacks fall into. Replay attacks, homophone attacks, and spoofing attacks all fall under this category. DolphinAttack [12] is a miscellaneous attack [1] since it does not attack the feature extraction or the underlying neural network, but rather the ability of microphones to pick up sounds that are otherwise inaudible to humans.

It is important to note adversarial attacks that have been done in the image domain, as there is potential for transferability. Methods such as the fast gradient sign method (FGSM), projected gradient descent (PGD), and the Carlini-Wagner (C & W) [13] attack have seen great success in computer vision, but it is not as effective in the audio domain due to the differences in how human vision and hearing work. For speech separation specifically, Takahashi et al. [14] specifically use FGSM and PGD in combination with a short-time power regularization to create an adversarial attack on Open-Unmix and Demucs, two systems designed for separating music. The attack is an untargeted attack, which means the only goal is to degrade the performance of the separation system.

Regarding adversarial attacks on automatic speech recognition, effective and imperceptible attacks have been developed. One such example is by Carlini and Wagner [15] who also created the previously stated Carlini-Wagner attack on images. It is a two-step attack. The first step is to find a perturbation using the connectionist-temporal

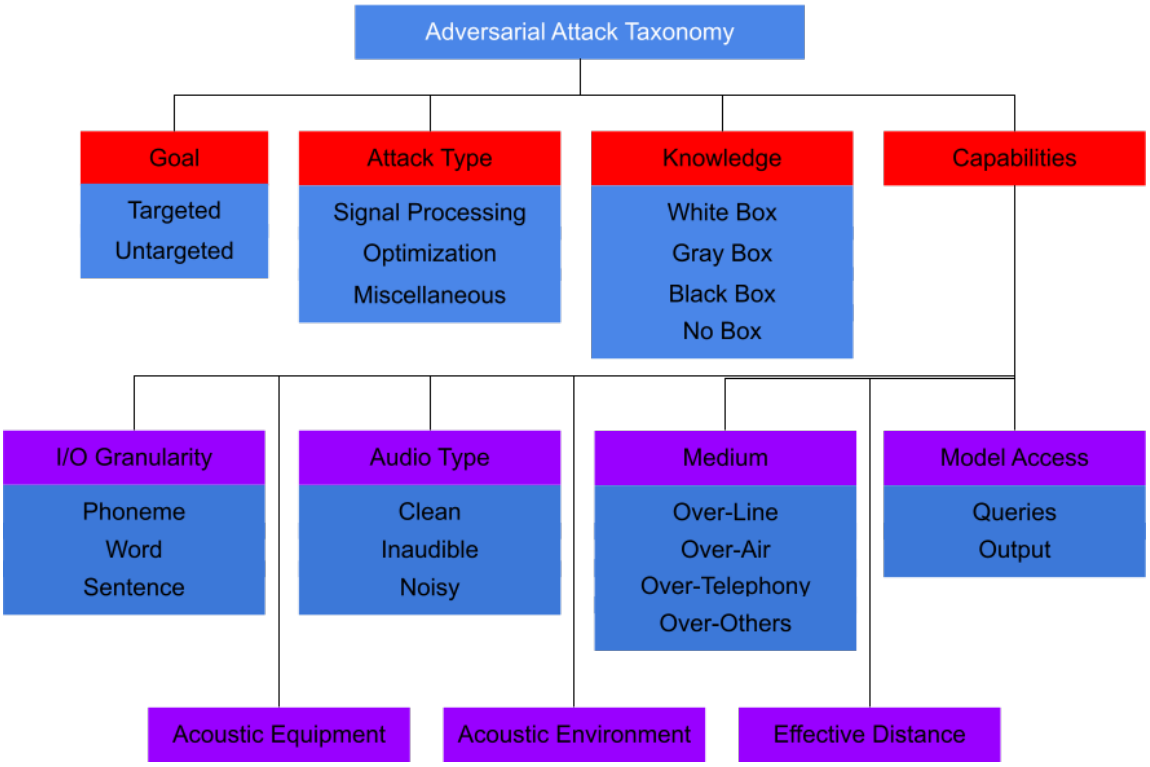

Fig. 1. Diagram showcasing Adversarial Taxonomy on VPS

classification (CTC) loss. The alignment determined by CTC loss is then fixed so a new perturbation is found using an improved loss function of Carlini and Wagner's design. With this, they produced "targeted adversarial examples with 100% success... with a mean perturbation of -31dB."

Another attack, designed by Schönherr et al. [16] utilized psycho-acoustic hiding to make the perturbations hidden. Similar to Carlini and Wagner's [15] work, the best alignment is calculated using forced alignment, which is then held fixed as backpropagation is applied to the audio sample. Hearing thresholds are also calculated for the sample and perturbation are then constrained to be within the imperceptible region of the audio sample. As a result, Schönherr et al. [16] are successful in creating an imperceptible attack 98% of the time.

3 PROBLEM STATEMENT

For this project specifically, we are investigating if there is a possibility of a targeted attack on speech separation. As previous work has shown, untargeted attacks are possible; however, they are the weaker attack type and only result in degraded performance of the system. Targeted attacks, on the other hand, are more interesting, and as a result of being the stronger type of attack, are more dangerous. In addition to the idea of a targeted adversarial attack, defending against such attacks would improve speech separation systems as a whole, making them more robust. As Takahashi et al. [14] found out, the black-box scenario greatly reduces the effectiveness of such attacks, meaning that there is no transferable attack. This is consistent with the findings of others, in which black-box attacks on VPS are not particularly prevalent in the current research.

3.1 Technical Challenges

The project introduces a few technical challenges and difficulties in its implementation. The first and most apparent challenge is that adversarial attacks have been used to great effect in the image-domain and this project is operating in the audio domain. Not only is the representation of audio and images different, but so are the mechanisms that drive human sight and hearing. For the most part, image data can be represented linearly, with pixel position and RGB data most commonly used to represent an image. Audio on the other hand is also inherently temporal, adding an extra dimension to consider when applying perturbation to the audio. The second challenge is that speech separation is not necessarily a classification problem. The challenge this poses is that most, if not all, adversarial attacks are designed with classification in mind. Therefore, adversarial attacks on speech separation have to be evaluated with different metrics than traditional methods like mean squared error, or cross entropy loss. It also differs from other established adversarial attacks because it is a one-to-many function without any set classes that an

output can be put into. This increases the difficulty of making a targeted attack greater, as there is not an easy way to target something when you do not know what to target.

4 SYSTEM DESCRIPTION

As stated before, adversarial samples are typically crafted for the image domain and therefore usually work best on models that work on images. Even though audio and image are different, we can apply similar methods to add noise to and input and cause a machine learning model to misclassify the sample, as demonstrated by Carlini and Wagner [15]. We add an ASR system to the pipeline to solve the problem of speech separation being a particularly difficult regression problem and turn it into a classification problem. Therefore, the inputs to the ASR will be the separated speech and the targets for a targeted adversarial attack is one of the separated sources.

4.1 Models Used

ConvTasNet was chosen as the speech separator of choice for the system because of its relative simplicity and smaller model size. Figure 2 is a high-level overview of the structure of ConvTasNet. It is also linear, making the computation simpler than models that use a logarithmic feature extraction. Wav2Vec2 is chosen as the ASR model of choice for similar reasons. Speech2Text differs from Wav2Vec2 in its feature extraction, which makes it an excellent target to test transferability of the attack. Instead of linear features, it uses mel-filter banks, which are logarithmic.

4.2 Gradient Descent

Gradient Descent (GD) is the standard method for many model training tasks, and it can be used to create adversarial samples as well. The perturbation is defined as:

$$\max_{\eta \in D} d(f(x+\eta), f(x)),\ D = \{n | C(n) < \delta\}. \tag{2}$$

x is the input audio, η is the perturbation, $d(\cdot,\ \cdot)$ is a chosen metric, C is some constraint and δ is a threshold. Typically, C can be the l_2 norm or l_{inf}. Using gradient descent, (2)

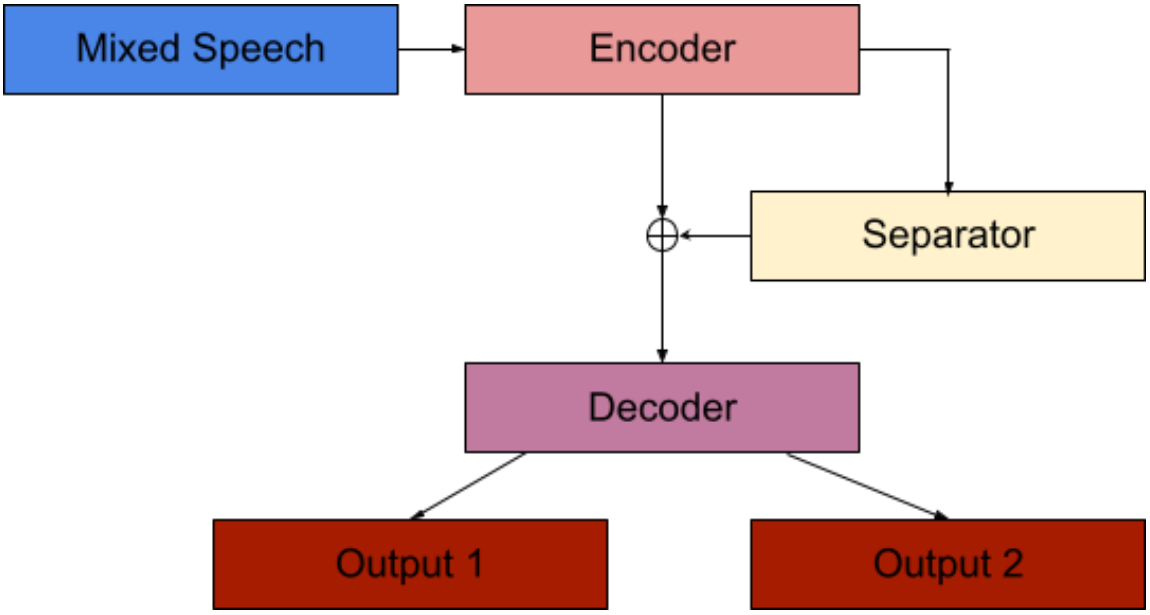

Fig. 2. Block diagram of ConvTasNet.

can be solved by minimizing the loss function L:

$$L(\eta) = -||f(x+\eta) - f(x)||_2^2 + \lambda C(\eta). \tag{3}$$

where λ is a scalar.

4.3 Fast Gradient Sign Method

FGSM [17], as it is named, is a quick method to calculate a perturbation. This perturbation is calculated as,

$$\delta = \varepsilon \text{sign}(\nabla_x L(f(x), y)) \tag{4}$$

where ε is the size of the perturbation, $L(f(x)\,y)$ is the loss function and y is the original, clean signal. This method is typically not targeted, but can be made targeted by changing y to the target signal, instead of the reference signal.

4.4 Projected Gradient Descent

Following FGSM, Projected Gradient Descent (PGD) is an extension of it, calculating the perturbation through the use of multiple steps instead of a single step. The idea behind PGD is to calculate it iteratively. After each step, project the perturbation back onto a defined ε-ball if the perturbation is too large. With x^t as the input signal at time step t, PGD is applied as:

$$x^{t+1} = \prod_{\varepsilon}(x^t + \alpha \text{sign}(\nabla_x L(f(x^t), \text{sg}(f(x)))) \tag{5}$$

where α is the step size, $\prod_{\varepsilon}$ is the projection to the ε-ball, and $\text{sg}(f(x))$ is the stopping gradient operation.

4.5 Proposed System

The core idea behind the system is to apply perturbations to the output of the speech separation system in the hopes that adding such a perturbation and replaying the result to the speech separation system would fool the system into mis-separating the sources. The system consists of ConvTasNet as the speech separation system and one of Wav2Vec2 or Speech2Text chained together such that the output of the speech separation model is the input of the speech recognition model. Error is calculated using connectionist temporal classification (CTC) loss and perturbations are optimized using Adam.

First, the mixed speech is sent through the speech separation system to get an idea of the "ordering" of the outputs, since the output is an array of shape [*batch*, *n*, *length*] where *n* is the number of outputs. In this case, the batch size is 1, so output[1, 0, *length*] is considered to be output number 1. Output number 2's provided transcription is therefore chosen to be the target transcription for the attack.

Adam is initialized on a tensor of zeros as long as the original mixture. That tensor is added onto the mixture and is fed through the speech separation system. Both outputs are then fed into the automatic speech recognition system. The labels used to calculate the

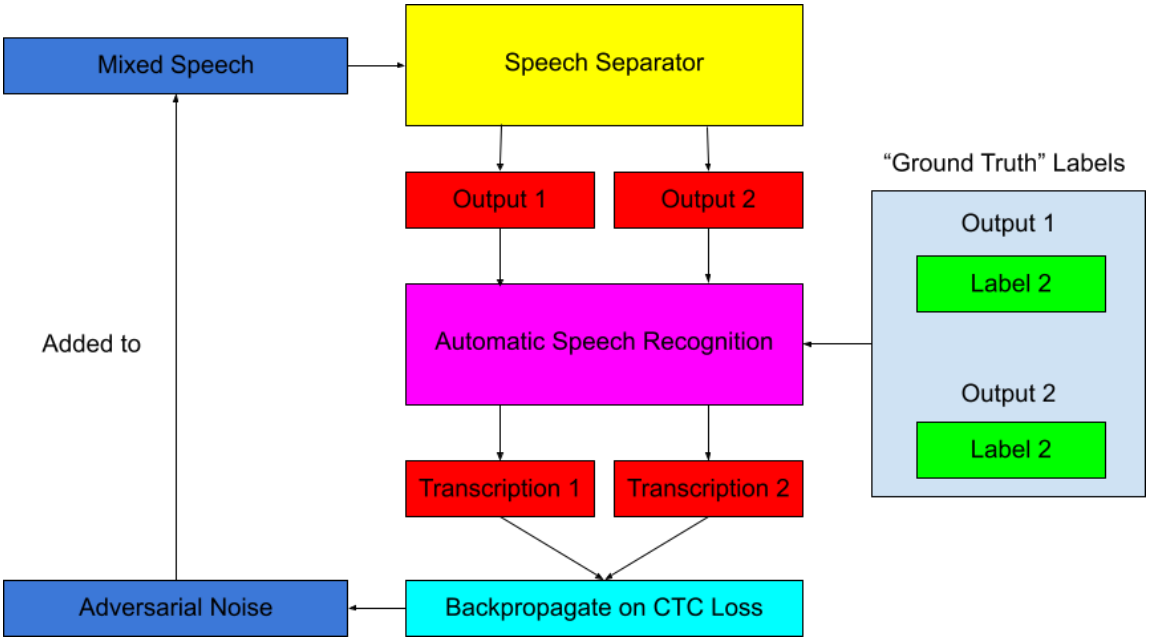

Fig. 3. Diagram of the full process of creating an adversarial sample. The speech separator used is ConvTasNet. The automatic speech recognition system is one of Wav2Vec2 or Speech2Text. The adversarial noise is calculated using one of unbounded gradient descent or projected gradient descent

CTC loss at the end of the process is the transcription of output number 2's twice. This helps calculate a delta that both maximizes the change in output 1 while keeping the resulting output's perturbations to a minimum for output 2. Figure 3 visualizes the process of creating an adversarial sample from mixed speech.

5 METHODOLOGY

5.1 Dataset, Software, and Hardware

The primary dataset used is Libri2Mix. This dataset is generated from Librispeech and has a multitude of options when creating the dataset, including the sample rate, whether to augment with the WHAM dataset [18] and the length of the sample (max being the longer of the two samples and min being the shorter). Adversarial samples were generated from the clean (no background noise), 16 kHz sample rate and max length subset of the dataset.

The primary language used in the development of the system is Python 3, written in PyCharm IDE on Ubuntu 20.04. Libraries used include Asteroid [6], HuggingFace Transformers [7], PyTorch and Torchaudio [19], librosa [20], numpy, pandas and jiwer.

Adversarial samples were generated on a computer using AMD Ryzen 5 3600 at 3.6 GHz and a founder's edition Nvidia GTX 3070 with 8 gigabytes of VRAM. Due to hardware limitations, only mixtures of length less than 7.5 seconds is used in the creation of adversarial samples. With more memory, longer mixtures can be used, but even with memory reduction methods such as gradient checkpointing and automatic mixed precision training, 7.5 seconds have been the maximum.

5.2 Metrics

This section describes the various metrics used to evaluate performance. The first two are considered speech separation metrics, as they can be used to evaluate the output of ConvTasNet. The latter metrics are evaluation metrics for ASR and are used to evaluate the efficacy of the attack.

5.2.1 Signal-to-Distortion-Ratio

Signal-to-Distortion Ratio (SDR) is a standard metric used to measure the quality of a sound sample. Let $\hat{s}$ be the estimate obtained from some algorithm. Typically, it is

calculated as:

$$\text{SDR} = 10\log_{10}\left(\frac{||s||^2}{||s-\hat{s}||^2}\right) \tag{6}$$

SDR is commonly used to evaluate the effectiveness of speech separation systems. In general, positive decibel values mean that there is more signal than there is distortion, and is desirable for a speech enhancement task like speech separation. A value of 0 dB would mean that the estimate is exactly the target and a negative dB means that the signal is degraded, as there is more noise than there is actual signal.

5.2.2 Degradation

Takahashi et al. [14] define three degradation metrics which are useful to quantify the strength of the perturbation and the strength of degradation in the performance of the speech separation system. These three metrics are degradation of separation DS_M, degradation of input DI_M and degradation of separation with additive adversarial noise DSA_M, based on some metric M. For this experiment M is SDR, to maintain parity with Takahashi et al. [14] for comparison purposes. These three metrics are calculated as follows:

$$\text{DS}_{\text{SDR}} = \text{SDR}(y, f(x)) - \text{SDR}(y, (f(x+\eta)) \tag{7}$$

$$\text{DI}_{\text{SDR}} = \text{SDR}(x, x+\eta) \tag{8}$$

$$\text{DSA}_{\text{SDR}} = \text{SDR}(y, f(x)) - \text{SDR}(y, f(x)+\eta). \tag{9}$$

Higher DS_{SDR} is a desirable result because it measures how much the SDR has degraded by the adversarial sample. The other two metrics measure the amount of adversarial noise. DI_{SDR} measures the SDR between the original and adversarial sample, while DSA_{SDR} measures the degradation of SDR when the adversarial noise is added to the outputs of the speech separation system instead of directly to the input.

5.2.3 Word Error Rate

On the side of the ASR, Word Error Rate is a standard metric used to evaluate the accuracy of the ASR through the number of deleted, inserted and substituted words in a sentence or phrase [21]. It is calculated as:

$$\text{WER} = \frac{S+D+I}{S+D+H}, \tag{10}$$

where S is the number of substitutions, D is the number of deletions, I is the number of insertions and H is the number of hits, or correct words. Substitutions, deletions and insertions are calculated using the Levenshtein distance, otherwise known as edit distance. Typically, a 5% word error rate is desirable, but to measure the success of the adversarial attack we want the word error rate to be higher. For this experiment, Word Error Rate is calculated with all output from the speech separation system. This means that in the most ideal situation where all the words for transcription 1 (corresponding to output 1) is wrong and all the words for transcription 2 are correct, the word error rate would be around 50% and can never be 100%. It is specifically used to measure the accuracy of the output transcription compared with the actual transcription.

5.2.4 Character Error Rate

Character Error Rate (CER) is a similar metric to Word Error Rate, in that it is calculated using the same formula as equation (10), but on the character level, instead of the word level. Character Error Rate is used to measure the difference between the transcript of adversarial output 1 and the transcript of adversarial output 2. In other words, it is how closely the two transcriptions match. Lower CER is desirable for the adversarial attack.

5.2.5 Match Error Rate

Match Error Rate (MER) aims to fix a problem with both WER and CER where the upper bound of WER and CER is not 1, but instead $\max(N_1, N_2/N_1)$ where N_1 is the

number of words in the input and N_2 is the number of words in the output. This is because if the amount of insertions exceeds the number of hits, then equation (10) will be greater than 1. Since this is not necessarily intuitive to the layman, MER provides an alternative that is bounded from $[0,1]$. MER is calculated as:

$$\text{MER} = \frac{S+D+I}{S+D+I+H} \tag{11}$$

and is simply the probability that a match from alignment is incorrect.

5.2.6 Word Information Lost and Word Information Preserved

Word Information Lost and its inverse Word Information Preserved are both based on a previously proposed metric for speech recognition systems called Relative Information Lost, or RIL [21]. RIL is calculated using Shannon entropy, which in information theory is the "average level of information" in a random variable's outcomes. RIL was not widely adopted because it is harder to calculate than WER, and any one-to-one alignment of input words to output words means that the RIL is zero. It measures how much word "information" is communicated by the output. WIL (and WIP) are approximations to RIL. Let $N_1 = H+S+D$ and $N_2 = H+S+I$. The formula for WIL is:

$$\text{WIP} = \frac{H}{N_1}\frac{H}{N_2} \tag{12}$$

and WIL is $1-WIP$. WIL and WIP both fix the drawback of RIL by removing the theoretical possibility of have zero information lost if there is a one-to-one I/O alignment of the texts.

5.3 Experimental Hyperparameters

The bulk of the samples generated were generated using gradient descent, however samples using projected gradient descent were also created and will be evaluated as well. FGSM, although mentioned, is more suited for untargeted attacks as it is one iteration of

gradient descent. For the Adam optimizer, the parameters are the default parameters from PyTorch (learning rate = 0.001, betas = (0.9, 0.999) and eps = 1e-8). For PGD, ε was chosen to be 0.2, while gradient descent's possible perturbations are unbound. Each sample underwent 500 iterations of gradient descent, as most of the samples converge by then. It takes approximately a minute to generate one sample, but with a more optimized training model it can be faster.

6 RESULTS AND ANALYSIS

In this section, we discuss the results on 1 speech separation system and 2 automatic speech recognition systems, as well as compare unbounded gradient descent and projected gradient descent. First, we discuss the results of the speech separation metrics and evaluate the amount of adversarial noise added to our input. Afterwards, we discuss the results of the attacks with automatic speech separation metrics to measure the success of the attack and if our goal is achieved. Finally, transferability of the attacks is discussed, using Speech2Text as the second automatic speech recognition system to test.

6.1 Speech Separation Metrics

To measure the performance of the speech separation portion of the system, we use two of the three degradation metrics introduced earlier, degradation of input and degradation of separation.

Table 1
Degradation Comparison

Attack	**DI_{SDR} (dB)**	**DS_{SDR} (dB)**
Takahashi et al. (l_2 norm reg.)	28.83	5.25
Takahashi et al. (STPF reg.)	30.33	5.23
Unbounded Gradient Descent	12.92	0.34
Projected Gradient Descent	6.27	0.29

It is important to remember that although both attacks are adversarial attacks, the proposed system is a targeted attack, compared to the untargeted goal of the other. Therefore, what can be considered a good result for one attack when degrading the performance of the speech separation can be considered poor for the other. In this case, a higher DI_{SDR} means that there is less adversarial noise and a higher DS_{SDR} means that there is more degradation in the output. The system achieves a DI_{SDR} of 12.92 dB and a DS_{SDR} of 0.34 dB with unbounded gradient descent and a DI_{SDR} of 6.27 dB and a DS_{SDR} of 0.29 dB with projected gradient descent. Compared to Takahashi et al.'s [14] attack,

this performs much worse in an untargeted situation, as it means that there is more adversarial noise and less degradation in the output of the system.

However, in the context of a targeted attack, a lower DS_{SDR} is needed for the automatic speech recognition to actually work and transcribe the output. DI_{SDR} can still be improved because less adversarial noise means that the attack is less perceptible to human ears. Takahashi et al.'s [14] attack is more effective in that regard, and the increased adversarial noise is a significant drawback of the proposed attack. Future work should be dedicated to improving this metric.

It is also interesting that projected gradient descent performs worse in all metrics compared to the unbounded gradient descent. Intuition would say that additional restrictions should reduce the amount of noise that is added to the sample. A possible explanation beyond the number of samples is that bounding the possible values that the perturbations can be causes less optimal solutions to be found. For example, say that an optimal perturbation at a certain sample is 0.3 dB. Since PGD bounds the perturbation, this must be projected onto the ε-ball and an alternate solution needs to be found. This difference could then be added to periods of silence, making the perturbations much more discernible and comparatively noisier.

6.2 Automatic Speech Recognition Metrics

WER, MER, WIL and WIP are all calculated using the jiwer library for Python.

6.2.1 Unbounded Gradient Descent

Table 2 shows the number of hits, substitutions, deletion and insertions calculated and used by jiwer to determine the four metrics in Table 3.

In Table 3, the WER of the adversarial sample exceeds the WER of clean samples by a difference of 60%. For the clean samples, MER and WER are extremely close, meaning that the number of insertions is extremely low compared to the number of hits, according to the formulas for both MER and WER. On the other hand, the adversarial samples'

Table 2
Hits, Substitutions, Deletions and Insertions

Type	Hits	Substitutions	Deletions	Insertions
Clean Samples	17962	663	83	44
Adversarial Samples (GD)	8475	8489	1744	1755

Table 3
Comparison of Clean versus Adversarial

Type	WER	MER	WIP	WIL
Clean Samples	4.22%	4.21%	92.38%	7.62%
Adversarial Samples (GD)	64.07%	58.58%	20.51%	79.49%

MER and WER differ by about 6%, which means that proportionally there are more insertions than there are hits.

If both transcriptions from outputs 1 and 2 of the speech separation system differ entirely, an ideal, perfect attack would have a WER of 52%. Expecting an error of 4% on the transcription of output 2 (which should be unmodified) and 100% on the transcription of output 1 would average out to be 52%. However, the WER is 64.07% indicating that there the attack has increased word error rate of the "correct" transcription as well. Closer inspection of the output transcriptions show that this is the case, and the desired result of the output transcriptions being nearly identical is still achieved.

With regard to both WIP and WIL, the information preserved is extremely low. It is expected to be low because the transcription is modified such that the original meaning is entirely different. Therefore, the results are as expected, and we can use it to measure the efficacy of the attack as effective. Similarly, since WIP and WIL are inverses, the "information" lost from this attack is high as the new transcriptions do not resemble the original at all.

Select transcriptions are provided in Table 4. Character Error Rate (CER) is used to measure the similarity between the "correct" transcription and the "adversarial" transcription. For this attack, the CER between the two output transcriptions from one adversarial sample is 24.77%. This generally means that about 1 in 4 characters in the transcription is incorrect. Calculating CER for the expected and correct transcriptions (not adversarial) gives us 114.1% CER, indicating that our attack "improves" the CER by a difference of nearly 90%. An ideal attack will have a far lower character error rate, but the lack of an appropriate baseline to compare against makes it difficult to make sure statements on the effectiveness. For the most part, the transcriptions are human readable, and recognized by the speech recognition system.

6.2.2 Projected Gradient Descent

Projected Gradient Descent, as mentioned in section IV, is gradient descent, but on each step, the result is projected back onto an ε-ball as an additional constraint on the optimization. The purpose of projecting is to make sure that the perturbations that are applied to the original sample are not too large and therefore not as noticeable. For this particular set of samples, 517 samples are created and are bounded within $\varepsilon = 0.2$ and projected with the L_{inf} norm. Table 5 shows the number of hits, substitutions, deletion and insertions in comparison to the same set of clean samples.

Compared against unbounded gradient descent, projected gradient descent reduced the amount of hits by 51.06% while unbounded gradient descent reduced the amount of hits by 52.81%. The difference is very small and not significant. An explanation could be that the unbounded gradient descent still is within the bounds of $\varepsilon = 0.2$ and only a small insignificant part of the waveform is actually projected back.

Table 6 shows the final calculations for WER, MER, WIP and WIL. For the most part, the results on the clean samples are around the same and is expected to be since this is a subset of the samples used to create the unbounded gradient descent samples. The results

Table 4
Select Transcriptions from Wav2Vec2

Mixture ID	Type	Transcription 1	Transcription 2
1580-141083-0016_61-70968-0024	Adversarial	I WAS IN SUCH A HURRY TO COME TO YOU YOU LEFT YOUR DOOR OPENX	I WAS IN SUCH A HURRY TO COME TO YOU YOU LEFT YOUR DOOR OPEN
	Clean	I WAS IN SUCH A HURRY TO COME TO YOU YOU LEFT YOUR DOOR OPEN	SPOKE THE SQUIRE LOSING ALL PATIENT AND IT WAS TO YOU THAT I GAVE ANOTHER PERSON CONSOLATION
237-126133-0008_2830-3979-0002	Adversarial	“ASKED PHRONSIE WITH HER LITTLE FACE CLOSE TO POLLY’S OWN¡unk¿”	“ASKED PHRONSIE WITH HER LITTLE FACE CLOSE TO POLLY’S OWN¡unk¿”
	Clean	“ASKED PHRONSIE WITH HER LITTLE FACE CLOSE TO POLLY’S OWN”	LET US BEGIN WITH THAT HIS COMMENTARY ONGULATIONS

on the adversarial samples are very similar as well, with a difference of 2.44%. This is similar for the other two metrics, MER and WIP.

However, the unbounded gradient descent still has a greater WER, so in terms of these specific metrics, the unbounded gradient descent performs better than the projected

Table 5
Hits, Substitutions, Deletions and Insertions

Type	Hits	Substitutions	Deletions	Insertions
Clean Samples	11527	464	52	27
Adversarial Samples (PGD)	5641	5315	1087	1020

Table 6
Comparison of Clean versus Adversarial

Type	WER	MER	WIP	WIL
Clean Samples	4.51%	4.5%	91.8%	8.2%
Adversarial Samples (PGD)	61.63%	56.82%	22.06%	77.94%

gradient descent. The difference is then the difficulty of the two problems in comparison to each other. Since projected gradient descent has additional constraints, it is computationally more expensive and harder to optimize. Observation of sample generation shows that this is the case, with sample generation taking around 90 seconds. It is also more intensive on GPU memory, explaining the fewer number of samples generated.

6.2.3 Transferability

The attack was further tested against Speech2Text (S2T) to evaluate the transferability of the attack. Table 7 compares the number of hits, substitutions, deletions and insertions, as calculated by jiwer.

Table 7
Hits, Substitutions, Deletions and Insertions

Model	Hits	Substitutions	Deletions	Insertions
Wav2Vec2	8475	8489	1744	1755
Speech2Text	5921	11366	1421	3030

Table 8
Comparison of Accuracy versus S2T and Wav2Vec2

Model	WER	MER	WIP	WIL	CER
Wav2Vec2	64.07%	58.58%	20.51%	79.49%	24.77%
Speech2Text	84.55%	72.77%	9.22%	90.78%	104.83%

Table 8 shows surprising results. The WER of the attack against S2T is higher than that of the attack against the model that is used to create the adversarial samples, but this is not necessarily correct. Closer inspection of the generated transcriptions, however, show that the targeted attack does not work against Speech2Text, despite the increased WER, MER and WIL. In this case, the word error rate is likely because of the introduction of noise to the output of the speech separation system, significantly degrading the performance of S2T in the transcription task.

A major difference between Wav2Vec2 and S2T is how features are extracted. S2T uses mel-filter bank features, while Wav2Vec2 operate directly on the amplitude of the waveform. It is possible that the mel-filter bank feature extraction operation is not robust to noise and thus fails to extract correct features for the separated speech. Here, the targeted attack failed to be transferred, but was unintentionally an untargeted attack against the speech separation system.

Another confirmation of the attack's non-transferability is the resulting CER of 104.83% when tested against S2T. A CER greater than 100% indicates that there are more insertions than there are hits and that there is a massive difference between the output transcript and the target transcript. If the attack was transferable, one would expect the resulting CER to be closer to that of the resulting CER from Wav2Vec2, which is 24.77%. In conjunction with the results from Table 8 the attack does not transfer when pitted against a different ASR system, but still heavily degrades the performance of that system.

7 CONCLUSIONS AND FUTURE WORK

In this project, we find that speech separation systems are susceptible to being used in an adversarial attack against automatic speech separation and are still vulnerable to adversarial attacks. We use Libri2Mix to create over 700 adversarial samples of varying efficacy at a sample rate of 16 kHz and show that the attack is relatively simple and has massive room for improvement and future research into defending against such attacks. It uses multiple queries to ConvTasNet and Wav2Vec2 to calculate the adversarial noise added to the original mixture, resulting in an overall 64% WER and a not insignificant amount of degradation of SDR. When tested against Speech2Text, the attack was found to be ineffective; however, it still degrades the performance of said model, achieving a WER of 84.55%. When comparing the two output transcriptions of ConvTasNet, the CER between the two was found to be 24.77%. While relatively high, it indicates that there is similarity between the two transcriptions, compared to the high CER of the clean transcriptions. We also find that the attack is not transferable, consistent with previous findings in the audio domain; however, it still significantly degrades the performance of other model in the untargeted setting.

While the proposed attack is nothing new, it leverages proven previous methods of creating adversarial samples in the image domain against speech separation tasks to produce incorrect transcriptions. The attack proves that adversarial noise can be added to a mixture of speech and survive the speech separation process, which can be dangerous when using speech separation to enhance speech for the purposes of creating a dataset, or production level speech recognition systems.

For future work, it is important to try the various defenses against adversarial attacks and test their efficacy in preventing mis-transcription. Another future investigation is testing speech separation models that have been trained specifically using datasets such as WHAM! which introduce ambient noise to the mixture. It would be interesting if the

mechanics behind separating ambient noise from speech would be similarly effective against adversarial noise. Most importantly, more work needs to be done to improve the DI_{SDR} metric. Finally, more experiments should be done to figure out if the targeted attack can become a chosen targeted attack, in which the attacker can choose the target transcription.

Literature Cited

[1] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. Traynor, "Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems," *CoRR*, vol. abs/2007.06622, 2020.

[2] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, "Continuous speech separation: Dataset and analysis," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7284–7288, 2020.

[3] J. Zhu, R. A. Yeh, and M. Hasegawa-Johnson, "Multi-decoder dprnn: Source separation for variable number of speakers," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3420–3424, 2021.

[4] Y. Luo and N. Mesgarani, "Conv-tasnet: Surpassing ideal time-frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, pp. 1256–1266, Aug 01, 2019.

[5] Y. Luo, Z. Chen, and T. Yoshioka, "Dual-path rnn: efficient long sequence modeling for time-domain single-channel speech separation," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 46–50, IEEE, 2020.

[6] M. Pariente, S. Cornell, J. Cosentino, S. Sivasankaran, E. Tzinis, J. Heitkaemper, M. Olvera, F.-R. Stöter, M. Hu, J. M. Martín-Doñas, D. Ditter, A. Frank, A. Deleforge, and E. Vincent, "Asteroid: the PyTorch-based audio source separation toolkit for researchers," in *Proc. Interspeech*, 2020.

[7] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, (Online), pp. 38–45, Association for Computational Linguistics, Oct. 2020.

[8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems* (H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, eds.), vol. 33, pp. 12449–12460, Curran Associates, Inc., 2020.

[9] C. Wang, Y. Tang, X. Ma, A. Wu, D. Okhonko, and J. M. Pino, "fairseq S2T: fast speech-to-text modeling with fairseq," *CoRR*, vol. abs/2010.05171, 2020.

[10] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," *CoRR*, vol. abs/1904.01038, 2019.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[12] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, (New York, NY, USA), p. 103–117, Association for Computing Machinery, 2017.

[13] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2017.

[14] N. Takahashi, S. Inoue, and Y. Mitsufuji, "Adversarial attacks on audio source separation," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 521–525, 2021.

[15] N. Carlini and D. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 1–7, 2018.

[16] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," *CoRR*, vol. abs/1808.05665, 2018.

[17] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv e-prints*, p. arXiv:1412.6572, Dec. 2014.

[18] J. Cosentino, M. Pariente, S. Cornell, A. Deleforge, and E. Vincent, "LibriMix: An Open-Source Dataset for Generalizable Speech Separation," *arXiv e-prints*, p. arXiv:2005.11262, May 2020.

[19] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), pp. 8024–8035, Curran Associates, Inc., 2019.

[20] B. McFee, A. Metsai, M. McVicar, S. Balke, C. Thomé, C. Raffel, F. Zalkow, A. Malek, Dana, K. Lee, O. Nieto, D. Ellis, J. Mason, E. Battenberg, S. Seyfarth, R. Yamamoto, viktorandreevichmorozov, K. Choi, J. Moore, R. Bittner, S. Hidaka, Z. Wei, nullmightybofo, A. Weiss, D. Hereñú, F.-R. Stöter, P. Friesch, M. Vollrath, T. Kim, and Thassilo, "librosa/librosa: 0.9.1," Feb. 2022.

[21] A. C. Morris, V. Maier, and P. Green, "From WER and RIL to MER and WIL: improved evaluation measures for connected speech recognition," in *Proc. Interspeech 2004*, pp. 2765–2768, 2004.

Appendix A

SELECT TRANSCRIPTIONS

This appendix contains examples of output transcriptions acquired from the adversarial samples generated by the attack. Mixture ID is the label given to the mix by the Libri2Mix dataset and is a combination of the ID given to the sample by LibriSpeech. Since it is impossible to predict the permutation output by ConvTasNet, both transcriptions and their possible matchings are given for each row.

A.1 Wav2Vec2 Transcriptions

The below table are some transcriptions chosen from the 700 plus samples generated for this project. It is impossible to list all transcriptions, due to length.

Table 9: Select Transcriptions

Mixture ID	Type	Transcription 1	Transcription 2
1580-141084-0027_5639-40744-0005	Adversarial	NO SIR CARTAINLY NOTQ	NO SIR CERTAINLY NOT
	Clean	NO SIR CERTAINLY NOT	FINALLY THE ONE PARTY WENT OFF EXULTING AND THE OTHER WAS LEFT IN DESOLATION AND WILL

Table 9: Select Transcriptions (cont.)

Mixture ID	Type	Transcription 1	Transcription 2
672-122797-0060_7729-102255-0035	Adversarial	IT IS A VERY STUPID STORY¡unk¿	IT IS A VERY STUPID STORY¡unk¿
	Clean	IT IS A VERY STUPID STORY	THE MILITARY FORCE PARTLY RABBLE PARTLY ORGANIZED HAD MEANWHILE MOVED INTO THE TOWN
4507-16021-0003_672-122797-0003	Adversarial	BUT THIS WAS WHAT THE TREE COULD NOT BEAR TO HEAR	BUT THIS WAS WHAT THE TREE COULD NOT BEAR TO HEAR¡unk¿
	Clean	SHE HAS A SON THEFT AND A DAUGHTER HUNGER	BUT THIS WAS WHAT THE TREE COULD NOT BEAR TO HEAR

Table 9: Select Transcriptions (cont.)

Mixture ID	Type	Transcription 1	Transcription 2
1320-122617-0017_6930-76324-0004	Adversarial	THEN AS IF SATISFIED OF THEIR LAFETY THE SCOUT LEFT HIS POSITON AND SLOWLY INTERED THE PLANET	THEN AS IF SATISFIED OF THEIR SAFETY THE SCOUT LEFT HIS POSITION AND SLOWLY ENTERED THE PLACE
	Clean	THEN AS IF SATISFIED OF THEIR SAFETY THE SCOUT LEFT HIS POSITION AND SLOWLY ENTERED THE PLACE	BUT JOYCE HAD NOT BEEN LISTENING ALL AT ONCE SHE PUT DOWN HER CANDLE ON THE TABLE AND FACED HER COMPANION

Table 9: Select Transcriptions (cont.)

Mixture ID	Type	Transcription 1	Transcription 2
121-121726-0006_61-70970-0003	Adversarial	HERE FOMGAVE THE BECAUSE OF ALL OUR FAULES	HEREDITY THE CAUSE OF ALL OUR FAULTS
	Clean	HEREDITE THE CAUSE OF ALL OUR FAULTS	IF FOR A WHIM YOU BEGGAR YOURSELF I CANNOT STAY YOU

Table 9: Select Transcriptions (cont.)

Mixture ID	Type	Transcription 1	Transcription 2
7729-102255-0023_237-134500-004	Adversarial	THEIR DISTINCTIVE CHARACTERS HOWEVER DISPLAY ONE BROAD AND UNFAILING DIFFERENCE¡s¿	THEIR DISTINCTIVE CHARACTERS HOWEVER DISPLAY ONE BROAD AND UNFAILING DIFFERENCE¡unk¿
	Clean	THEIR DISTINCTIVE CHARACTERS HOWEVER DISPLAY ONE BROAD AND UNFAILING DEFERENCE	I"I PRAY FOR YOU BUT THAT'S NOT THE SAME AS IF YOU PRAYED YOURSELF"

A.1.1 Speech2Text Transcriptions

Below are the same samples from above, but from Speech2Text instead.

Table 10: Select S2T Transcriptions

Mixture ID	**Type**	**Transcription 1**	**Transcription 2**
1580-141084-0027_5639-40744-0005	Adversarial	finally it was unpardonable in althegie and the others of letters from desolation and awoke	no sir certainly not miss
	Clean	knoweth her shaking me not	finally the one party went off exulting and the other was left in desolation and woe
672-122797-0060_7729-102255-0035	Adversarial	the military fort carlisle's private organize had meanwhile moved into the town	it is a very stupid story
	Clean	it is a very stupid story	the military force partly rabble partly organized had meanwhile moved into the town

Table 10: Select S2T Transcriptions (cont.)

Mixture ID	Type	Transcription 1	Transcription 2
4507-16021-0003_672-122797-0003	Adversarial	it has no swarm yes it's three o'clock our two soldiers	but this was what the tree could not bear to hear
	Clean	she has a son theft and a daughter hunger	but this was what the tree could not bear to hear
1320-122617-0017_6930-76324-0004	Adversarial	but anyway it had not been listening all at once to keith's quick eyes podded along the table in a picture of her companion	then as if satisfied of their safety the shout left his position and slowly entered the flames
	Clean	then as if satisfied of their safety the scout left his position and slowly entered the place	but joyce had not been listening all at once she put on her candle on the table and faced her companion

Table 10: Select S2T Transcriptions (cont.)

Mixture ID	Type	Transcription 1	Transcription 2
121-121726-0006_61-70970-0003	Adversarial	if only when you're back in yourself i've had enough to use	credity the cause of all our faults
	Clean	heredity the cause of all our faults	if furwin you beggar yourself i cannot stay you
7729-102255-0023_237-134500-004	Adversarial	high places are you but that's not the thing i think very good feelings devotive	their distincting characters however displays one broad and unfailing difference
	Clean	their distinctive characters however display one broad and unfailing difference	i pray for you but that's not the famous if you pray yourself