Audio Adversarial Attack Analysis

Xinyi Liu, Poornima Haridas, Priyank Pathak
Department of Computer Science, New York University, New York, NY 10012, USA
xinyi.liu@nyu.edu, ph1391@nyu.edu, ppriyank@nyu.edu

Abstract

In this project we aim to construct robust adversarial audio attacks on automatic speech recognition systems. We start with basic perturbations of the sound waveform and then move on to adversarial attacks. Given any audio waveform, we then apply our targeted black-box iterative optimization-based attack to IBM Cloud Speech-To-Text APIs and Google Cloud Speech-To-Text APIs. We are able to successfully attack the APIs by changing most of the original transcription. Our source code is publicly available on GitHub[1] (Demo). [2]

Introduction

Adversarial attacks are meant to fool Machine learning models which ideally should be disguised and hidden from human perception. Although image based adversarial attacks are a very hot topic of research and have been around since 2014, audio adversarial attacks are a relatively less explored field. The attacks are mostly designed to fool machine learning models of automatic speech recognition systems (ASR) or Speech-To-Text models. ASR is an interdisciplinary subfield of computational linguistics that develops methodologies and technologies that enables the recognition and translation/transcription of spoken language into text by computers (speech to text).

Audio Adversarial attacks are much harder to generate compared to hiding noise in images. A convincing attack should be unperceived by human ears with the help of frequency masking (increasing the noise frequency below human threshold of hearing or masking the noise in neighbouring wavelet patterns).

Adversarial attacks can be classified based on how specifically they attack a model's output into the following:-

1. Targeted attacks - Given any natural waveform x, we are able to construct a perturbation δ that is nearly inaudible but so that $x + \delta$ is recognized as any desired phrase. In this work we aim to achieve a targeted attack.
2. Non-targeted attack - This is a relatively easy attack. It includes basic audio manipulation techniques including volume change, speed tuning, pitch modification and combination of multiple audio clips. This is generally enough to confuse the model that then outputs gibberish.

Attacks can also be classified based on how they are designed:-

1. White-Box attack - The specifics of the ASR model is known to the attacker and the attacker uses the same to malign the output.
2. Black-Box attack - The only input available to the attacker is outputs from samples passed to the model by the attacker(active research topic). The ASR models provided by common cloud APIs are black-box and we formulate a black-box attack on the same.

Recently there has been a surge in the advancement of robust attacks, where the end goal is to make the attacks immune to the medium of communication of the attack. Most times attacks don't transmit across a medium (screenshot of image attacks, or re-recording of the audio signals). In this paper we have focused on creating a black-box audio adversarial attack which is also robust such that it transmits through a medium (air in this case).

Problem Formulation & Cloud API Survey

The focus of our project was producing robust attacks and explore the field of audio adversarial attacks. These topics were decided after an extensive literature survey (detailed in the next section) where we realised that there was a lack of audio adversarial attacks and also less research in the field of robust attacks in any domain. The field of audio attacks is nascent and it was chosen to be able to make a significant contribution to the same.

To narrow down the type of task (image, OCR, ASR etc) and Cloud service (Google IBM, AWS, Azure) to be targeted, we conducted an extensive survey of various cloud APIs and the services they provide along with their free-trial limitations. The cloud platforms surveyed were:-

1. Google Cloud Platform
2. IBM
3. Azure
4. AWS

[1] https://github.com/ppriyank/Adveserial-Attacks

[2] https://github.com/ppriyank/Adveserial-Attacks/tree/master/demo

5. Clarifai

The final decision was made once we narrowed down on the reference papers and based on the ease of providing input, ease of setting up and working with the API and the free-trial offers. Therefore, we chose to attack the Google Cloud Speech-To-Text API(60 min/month for free) and IBM Cloud Speech-To-Text API (500 min/month for free). Since IBM had more mins/month free, we tuned our model based on results of the IBM model.

Related Work

The field of audio adversarial attacks is a nascent field that became significantly popular after Carlini and Wagner's work in 2018. In this section we detail our literature survey and explain in detail the papers that we chose to modify and replicate.

The **Deep Speech model** (Hannun et al. 2014) is a simple end-to-end speech recognition system (Fig 1). It is the most commonly attacked model for all adversarial attacks in the domain. Our reference papers focus on attacking the same and we thus describe this model briefly in our report.

It comprises of a relatively simple architecture. A recurrent neural network (RNN) is trained to ingest speech spectrograms and generate english text transcriptions. Fig. 1 shows the architecture which accepts $x(i)$, a time-series of length $T(i)$ where every time-slice is a vector of audio features, $x(i)_t, t = 1, ..., T(i)$ and C, a set of context frames. The output of the 1st three layers is - $h(l)^t = g(W^l h_t^{l1} + b^l)$, where $g()$ is a clipped rectified-linear activation function (ReLu) and W^l, b^l are the weight matrix and bias parameters for each layer.

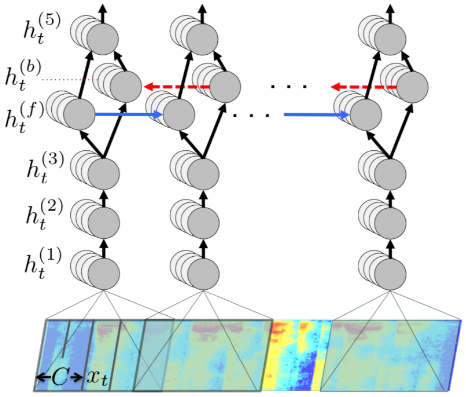

Figure 1: Visualization of Deep Speech model taken from (Hannun et al. 2014)

Carlini and Wagner's work was the first of its kind to do adversarial attack on audio signals. They essentially designed a white box attack on Deep Speech pre-trained model. Their attack however, was limited to white box approach. They did not explore black box attacks or robustness of an attack. The approach is a classic example of targeted white box attack, where the noise is trained using backpropagation and gradients of known model to create noise to force the output of the model to produce the desired output.

Given an original normalized signal x, and a trainable parameter δ the following steps were applied by the paper:-

1. Short Time Fourier Transform (STFT) of the input signal $(x + \delta)$, with approximations to make this step completely differentiable, as the gradients will be passing through to train this noise.
2. Weights of the model (Deep Speech) are frozen to prevent any training.
3. Limit the amount of noise added to minimize human hearing, by minimizing the relative decibels of the noise added with respect to the original signal.
4. Generalizing the noise by introducing a random Gaussian noise in addition to noise

The paper introduced a decibel loss and used CTC loss (Graves et al. 2006) to force the noise to get the desired target phrase. Decibel loss limits the amount of perturbation $DB_x(\delta)$. Further $L2$ norm is added as the one the major loss terms.

$$DB_x(\delta) = DB(\delta) - DB(x) \tag{1}$$

where $DB(x) = max_i \; 20 \; log_{10}(x_i)$

Training is done in a 2 step process:-

1. The decibel loss is reduced with a desired range τ and then CTC loss is minimized till a local minima is reached.
2. Next τ is further reduced and the cycle is repeated.

Qin et al.'s work build upon Carlini and Wagner's work in order to make it robust. Similar to previous approaches its an example of a targeted white box attack on lingvo model (Shen et al. 2019) which is a 2019 SOTA for speech to text transcription. The entire training takes place in two stages, one for modified training of Carlini and Wagner and one for robustness of the attack. The paper introduces normalized PSD loss to mask the noise in the neighbouring wavelet patterns (by minimizing spectral energy density of the noise).

$$l_\theta(x, \delta) = \frac{1}{\lfloor \frac{N}{2} \rfloor + 1} \sum_{k=0}^{\lfloor \frac{N}{2} \rfloor} (max(p_\delta(k) - \bar{}\theta_x(k), 0) \tag{2}$$

$$p_\delta(k) = 96 - max_k p_x(k) - p_\delta(k) \tag{3}$$

where $p_\delta(k) = 10 log_{10} |\frac{1}{N} s_x(k)|^2$ which is a 2-Dimension power density matrix having $\lfloor \frac{N}{2} \rfloor$ rows.

The paper suggested and implemented the following modifications:-

1. Instead of decibel noise, the model clips value to a predefined range τ ensuring decibel noise is always well within the given range.
2. The only loss for the step 1 training is CTC loss, that is to get the desired phrase with the requisite target phrase.
3. For the step 2, no bound on noise is applied while the model is tasked to minimized the loss $CTC + \alpha * l_\theta(x, \delta)$
4. The value of alpha is adaptive and is selected using adaptive boosting, depending upon the value of CTC loss.

5. Stage 2 uses a large number of Room impulse responses **r** which are based on various room configurations (the room dimension, source audio and target microphone location, and reverberation time). This function r when convoluted with $x + \delta$ creates a perturbed signal helping to train the noise in various room environments, with an eventual goal to be transmitted across the room unperturbed (Fig.2).

In this project, we apply Qin et al. and Carlini and Wagner's attacks to the cloud services and also modify Carlini and Wagner to make it more robust.

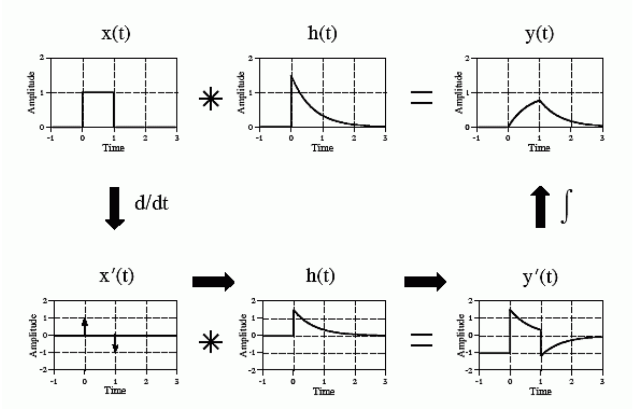

Figure 2: Visualization of convolution on signals taken from (Smith 2019)

Methodology

Robust Black Box Attack

We modified Carlini and Wagner's approach to make it eligible for black box attack by observing the behavior of IBM's cloud to modify the loss functions [3]. We summarize these modifications as follows :

1. Use L2 normalization in addition to whitening (0 mean and 1 standard deviation normalization). This will lower the audibility of the original sound by a factor of around 10.

2. Keep a ratio of 50:1 of original sound vs added noise. Hence we used a combination of : $audio(original) * 5 + 0.1 * noise(trainable)$ (seems to work for IBM Cloud).

3. We rarely use decibel loss as such a loss only effects one of the coordinates (the one with maximum value). Instead we focus more on L-2 norm which we found to have most overall impact.

4.

$$Loss = \begin{cases} 10 * CTC_{weight} * CTC_{loss} + \\ 0.005 * (DB_x(\delta) - tau) + \\ l2_{weight} * l2_{norm}, & \text{if } DB_x(\delta) < tau \\ \\ CTC_{weight} * CTC_{loss} + \\ 10 * 0.005 * (DB_x(\delta) - tau) + \\ 10 * l2_{weight} * l2_{norm}, & \text{otherwise} \end{cases} \quad (4)$$

5. We use adaptive boosting to tune the weights of various loss functions ($l2_{weight}$, CTC_{weight}). If the CTC loss is well below a predefined threshold, its weight changed to $max(CTC_{weight} - 0.1, 0.05)$ while $l2_{weight}$ is given a boost of 0.2. At the start of every iteration, CTC_{weight} is reset to initial weight, while τ is subtracted with 3.

One notable observations is that its was easy to get the desired targeted phrase but very difficult to reduce the audibility of the introduced noise.

Evaluation Metrics

We evaluate our results based on popular metrics that measure the difference between a reference and generated sentence. The metrics we used are listed below:-

1. Recall-Oriented Understudy for Gisting Evaluation 2 (ROUGE -2) Scores - We use the rouge library in python to calculate the same.

$$Recall = \frac{number\_of\_overlapping\_2\_gram_words}{total\_2\_grams\_words\_in\_reference\_summary}$$

2.

$$Precision = \frac{number\_of\_overlapping\_2\_gram\_words}{total\_2\_grams\_words\_in\_system\_summary}$$

3. F1 score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

$$F1 = \frac{2 * (Recall * Precision)}{(Recall + Precision)}$$

4. BLEU Scores (Bilingual Evaluation Understudy Score) - Evaluation metric for given two sentences proposed by Papineni et al.. BLEU's evaluation system requires two inputs: -

   (a) A numerical translation closeness metric, which is then assigned and measured against

   (b) A corpus of human reference translations

   We use the BLEU metric calculation provided in the NLTK library.

We use F1 score as the more prominent score across the evaluation as that is the most recognized and robust metric.

[3] https://github.com/ppriyank/Adveserial-Attacks/blob/master/paper2018/main3.py

Results

Non-targeted Attacks

1. **Long audio:** Audio files with longer length (> 20s) are fed to the cloud text-to-speech models. Table 1 presents the results of a 22 seconds audio clip. IBM Cloud Speech-to-Text has a better performance compared to Google service. Google service is transcribes "*the accepted*" as "*except*" as well as misses "*thought that creativity occurs on the right side of the brain, and the arts*" from the middle of the original transcription. This resulted in a low F1 score. Thus, for real-time cloud transcription models, longer audios successfully fool the models in most cases.
2. **Volume:** We increased and decreased the volume of the clip separately and tested the results. From Table 2, we can see that Google does not recognize audios that are too quiet or too loud. The original audio file has an F1 score of 0.878 on IBM. Lowering the volume by 50 dB, forced the IBM model to wrongly recognize some hard words like "*timaeus of plato*" as "*to me is way too*". Increasing the volume by 50 dB, resulted in IBM returning completely different words. This may be because some noises are enlarged during the process or that the models were trained on very limited and clean audio clips.
3. **Speed:** Table 2 shows that IBM is making mistakes on words with similar pronunciation when the speed is reduced. For example, "*hence*" is recognized as "*hens*", and "*poem*" as "*pool*". Hard words like "*timaeus of plato*" are hard to be output correctly for both slower and faster scenario. In IBM generated transcripts, "*%HESITATION*" are - Uh, Um, Ah, etc. Google benefits from the reduction in speed with a higher F1 score, while outputs a random word "*Flatow*" when the speed is increased and gets a lower F1 score.
4. **Pitch:** We further tried to change the pitch of the audio clip (see Table 2). Though humans can very well recognize the content in the speech, IBM model returns a completely wrong transcription, while Google returns no output. This again points towards clean traning dataset.
5. **Non-targeted noise:** We finally tried to add music and band limited noise as the background of the audio. As shown in Table 2, IBM model slightly under performs on hard words like "*timaeus*" and "*Lucretius*".

Targeted Attacks

Since the free speech-to-text services quota is limited, we are able to experiment with very limited audio files. In this report, we use the same short audio to illustrate our results for the non-targeted attacks.

We set the target phrase as "*this is a cml course*". Table 3 (original paper) shows that the F1 score drops when being fed the adversarial audio generated by Carlini and Wagner's model, but when we record that and send to the model again, the F1 score eventually goes up. This shows that this adversarial attack may slightly disturb these cloud APIs, but the adversarial information contained is not robust enough after the attacked sound transfers through air (the medium in our case).

Table 4 shows the outcome for modified version of Carlini and Wagner's work (our proposed methodology). Our targeted attack caused a significant drop in F1 scores of both the models. A recorded version of the the adversarially attacked audio caused a slight decrease (favourable) in Google's F1 score and a slight increase IBM's F1 score (unfavorable). The F1 scores of the recordings of the attacked audios, remain within 0.05 of the unrecorded (w/attack in the table) audios, which shows robustness. It also performs much better (in this case, able to reduce/maintain the F1 score even after re-recording) than the original paper (table 3), which has significant increases in the F1 scores after testing a recorded version of the attack (especially on Google's model). The slight increase of F1 score by our model, shows scope for future improvement to increase the robustness.

For Qin et al.'s work, we set the target as "*this is a cloud machine learning course*" when applying it to Shen et al.'s Lingvo system. Given 100 simulated room impulse response generated by pyroomacoustics, the generated adversarial example gets an overall success rate of 52.8% on the local white box Lingvo system (success here means the model gives exactly the same output as the target introduced by the attacker). We further apply these adversarial audio clips and their recorded-in-air version to cloud services. Table 5 shows that the robustness reduces in real environment.

During the above experiments with different models, we have tried multiple target phrases like "*this is a test*", "*my name is ... from NYU*", etc. We surprisingly find that the target we set and the new words appearing in the transcripts generated by the cloud models are correlated. We also tried to investigate the possible mapping, intending to generate more ribust targeted black box attack on cloud ASR models. There are two possible ways to perform mapping:

1. Replace all characters in original transcription by a desired letter,
2. Set the target as the original transcription appended characters like "aaaa...aa", "bbbb...bb" etc.

The first method may work because we are canceling the original signal as well as introducing the desired characters. The second method seems better to us since we no longer need to cancel the original sound. This was not possible though, since the calculation of CTC loss is based on the assumption: the new target phrase has a shorter length than the original transcript. Therefore, we tried the first method with character "a" to "f". We removed the words in the original script from the output of IBM cloud audio recognition models, and collected frequently appearing words for each target phrase. Below we populate the words that most occurred when we run our experiments with different alphabets:-

- a: till, with
- b: flames, female, place, clean
- c: (random words)
- d: fourth, crew, please, later
- e: premier, arrive, wait, female
- f: four, list, later, please

Original	*The accepted definition of creativity is production of something original and useful, and it is commonly thought that creativity occurs on the right side of the brain, and the arts play an important role in enhancing it. But according to a new research, creativity isn't about freedom from concrete facts.*	F1=1
IBM	*the accepted definition of creativity is production of something original and useful and it is commonly thought that creativity occurs on the right side of the brain and the arts play an important role in enhancing at but according to new research creativity isn't about freedom from concrete facts*	F1=0.928
Google	*except the definition of creativity is production of something original and useful and it is commonly occurs on the play an important role in enhancing according to new research creativity isn't about Freedom From Concrete facts*	F1=0.690

Table 1: Results of IBM and Google Cloud API for long audios

This is points to the existence of some mapping patterns. In future work, efficient targeted attacks with limited number of queries on cloud platforms can be designed if all or most mapping sequences can be isolated.

Contribution / Participation

Our Novel Contribution

1. The modification of Carlini and Wagner, which is a white-box attack on Deep Speech, to make the more robust attacks that work on the black-box models of Google and IBM cloud ARS services.
2. The application of basic but effective attacks. These were easy to implement, but produced effective results on both the cloud platforms. The results are shown in 2.
3. We surveyed available Cloud APIs and all their services and listed down the free-trail specifications of each service.
4. We studied the patterns that developed in the outputs being produced by the cloud services to try and enhance future black-box attacks.
5. Our model outperforms (Carlini and Wagner 2018) model and is more robust when compared to it (refer to Table 3 and Table 4, with a detailed anaylsis in the the result section)

Individual Contribution

1. Xinyi Liu - Testing IBM Cloud API, OCR model attack, implemented basic attacks, applying the Qin et al. attack, Cloud API survey.
2. Priyank Pathak - Literature review, modification of Carlini and Wagner to make it robust, applying the Qin et al. attack, OCR model attack.
3. Poornima Haridas - Literature review, testing Google Cloud API, applying Guo et al. attack, applying Taori et al. attack, Cloud API survey.

Trials

We tried several other attacks in the audio and image domains. These were either unsuccessful or could not be completed due to lack of time.

1. Chen and Jordan introduced a boundary shift based attack called HopSkipJump attack. We used their attack on OCR images to change the output of the image. The results of the same can be seen in Figure 3.
2. Taori et al.'s work was identified as a project that demonstrated a black-box attack on the Deep Speech model. Since the output of the Deep Speech model is more accessible and iterable, it worked well on the Deep Speech model, as claimed by the authors, but it did not work as well on the Google and IBM API, black-boxes. It was fine-tuned to attack only the Deep Speech model. It also lacked robustness.
3. Guo et al. demonstrated a relatively simple and query restricted black-box attack on the Google Cloud Vision API. We tried to replicate the results, but were unsuccessful as the model was very slow to train an image. Their attack on the Google Cloud API required too many GCP credits and therefore we could not attempt the same.

Questions and Answers

Here we address the questions put forward to us in the class:-

1. Can we use audio perturbations to make audio signals clearer?

   Answer: Yes, audio perturbations can be used to make audio signals clearer. A few examples of improving audio signals with perturbations could be:-

   (a) Introducing new noise signals that cancel the existing background/white noise.

   (b) Increasing the volume.
2. Can songs be attacked?

Original	*and hence we find the same sort of clumsiness in the timaeus of Plato which characterizes the philosophical poem of Lucretius*	F1=1
IBM (original)	*and hence we find the same sort of clumsiness in the two mayors of Plato which characterizes the philosophical poem of Lucretius*	F1=0.878
Google (original)	*and hence we find the same sort of clumsiness in the Tamia's of Plato which characterizes the philosophical poem of lucretius*	F1=0.899
IBM (-50DB)	*we find the same sort of clumsiness the to me is way too which characterizes the philosophical form of recreation*	F1=0.462
Google (-50DB)	*No Output*	F1=NA
IBM (+50DB)	*this this is a free service*	F1=0.000
Google (+50DB)	*No Output*	F1=NA
IBM (0.5x)	*and hens we find the same sort of clumsiness in the to me is the way to which characterizes the philosophical pool one of Lucretius*	F1=0.500
Google (0.5x)	*and hence we find the same sort of clumsiness in the Timaeus of Plato which characterizes the philosophical poem of lucretius*	F1=0.999
IBM (1.5x)	*and hence we find the same sort of clumsiness the today as of late %HESITATION which characterizes the philosophical poem of Lucretius*	F1=0.634
Google (1.5x)	*and hence we find the same sort of clumsiness in the tourney is a Flatow which characterizes if it was awful poem of lucretius*	F1=0.605
IBM (pitch change)	*yes so it was yes right right those awful problem*	F1=0.000
Google (pitch change)	*No Output*	F1=NA
IBM (band limited noise)	*and hence we find the same sort of clumsiness in the two mayors of Plato which characterizes the philosophical poem of Lucretius*	F1=0.732
Google (band limited noise)	*and hence we find the same sort of clumsiness in the Tamia's of Plato which characterizes the philosophical poem of lucretius*	F1=0.899
IBM (music - 30 dB)	*and hence we find the same sort of clumsiness in the two main areas of Plato which characterizes the philosophical poem of the creation*	F1=0.698
Google (music - 30 dB)	*and hence we find the same sort of clumsiness in the Tamia's of Plato which characterizes the philosophical poem of lucretius*	F1=0.899

Table 2: Results of IBM and Google Cloud API for attack involving volume changes, speed changes, pitch changes and noise addition. Noise added is band limited noise and adding music (music volume reduced by 30 dB) in the background

IBM (w/ attack)	*and hence we find the same sort of clumsiness in the to me as Plat which characterizes the philosophical form of recreation*	F1=0.683
IBM (recorded)	*and hence we find the same sort of clumsiness the to me as a way to %HESITATION which characterizes the philosophical form of recreation*	F1=0.512
Google (w/ attack)	*and how to recline the same sort of clumsiness in the team is Plato's which characterizes the clothes off*	F1=0.421
Google (recorded)	*and hence we find the same sort of clumsiness in the tomatoes of Plato's which characterizes the philosophical poem of lucretius*	F1=0.799

Table 3: Results of IBM and Google Cloud API for targeted attack: Carlini and Wagner's work

IBM (w/ attack)	*Hey with fish finder things sort of close the two layers of clay to characterize it was awful for*	F1=0.053
IBM (recorded)	*find things sort of close the clear way to characterize the philosophical issue*	F1=0.125
Google (w/ attack)	*required the same sort of clumsiness in the two layers of Plato which characterizes the clothes off*	F1=0.555
Google (recorded)	*replying the same sort of clumsiness in the tomatoes with Plato which characterizes the clothes off*	F1=0.514

Table 4: Results of IBM and Google Cloud API for targeted attack: our modified version based on Carlini and Wagner's work

IBM (w/ attack)	*it's a crime scene the wait*	F1=0.000
IBM (recorded)	*it's a class thing*	F1=0.000
Google (w/ attack)	*Honda Kingsport*	F1=0.000
Google (recorded)	*N/A*	/

Table 5: Results of IBM and Google Cloud API for targeted attack: Qin et al.'s work

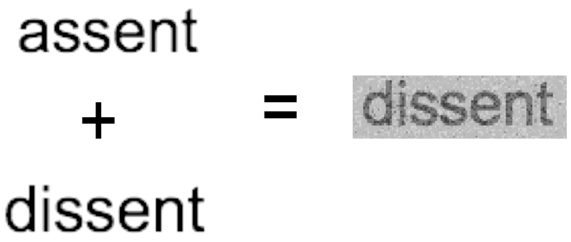

Figure 3: Resulting image after HopSkipJump attack. Target (hidden message is "assent") and initial image look like "dissent". Output of the tesseract model is "assent". Upon the screenshot (test for robustness) the model output is "**cissent**".

Answer: Music is much more complex to attack. The frequency and extension of words in various songs make it much more difficult to attack. Moreover, there are no pretrained models that take music as an input and therefore there were no reference points for us in the domain.

Insights

1. Break longer sentences to smaller ones, to be able to compare and attack the APIs, as they perform better on shorter audio files.
2. Producing targeted white-box attack is much easier than producing black-box attacks since the CTC loss can converge easily, easily producing the desired target phrase. This fails for black-box models.
3. Reducing the noise to a level such that it is imperceptible to human ears is the toughest challenge. The ratio of original signal and adversarially attacked audio, matters a lot. In our trials any ratio greater than 50:1 hampered the attack (50 × original signal + 0.1 × noise).
4. Patterns identified in the attacked transcription can be used to tune the attacks and make them stronger.
5. Changing the pitch has a huge impact on the F1 score. This shows data bias in the training of the cloud models (possibly female voices).
6. The models often get confused when it they encounter words with similar pronunciation.
7. The models also struggle with large words and spilt them into smaller more common words. This means that they give higher probability to common dictionary words ("*timaeus*" is often recognized as "*two mayors*", "*team*" and "*Tamia's*").
8. To produce robust attacks we need to apply various room impulse response functions (**r**) to the audio before training with the noise.
9. The failure of Google API on basic perturbations shows bias towards training data that may be relatively clean and uniform.
10. Both IBM and Google cloud services definitely don't use either Deep Speech or Lingvo models (or use some other model in ensemble).

Conclusion

In this project we implemented and applied various adversarial attacks with a focus on making all attacks robust. We implemented a new robust audio adversarial attack that is able to successfully reduce the F1 score of the transcripts generated by both IBM and Google Cloud Speech-To-text services. We also successfully attacked both the APIs with basic audio attacks (introducing unintelligent noise, varying frequency etc). Our results show that the IBM Cloud Speech-To-Text service seems to be more robust to basic attacks while the Google API is more robust to the advanced attacks. Finally, we tried to evaluate robust image attacks by applying available attacks from research to the Black-box models of Google and IBM Cloud Vision APIs. Our future work includes Black Box targeted attack on cloud ASR by discovering the mapping pattern between outputs of cloud APIs and our target, and further investigation on robustness of image adversarial attacks.

References

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

Chen, J., and Jordan, M. I. 2019. Boundary attack++: Query-efficient decision-based adversarial attack. *arXiv preprint arXiv:1904.02144*.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, 369–376. New York, NY, USA: ACM.

Guo, C.; Gardner, J. R.; You, Y.; Wilson, A. G.; and Weinberger, K. Q. 2019. Simple black-box adversarial attacks. *arXiv preprint arXiv:1905.07121*.

Hannun, A.; Case, C.; Casper, J.; Catanzaro, B.; Diamos, G.; Elsen, E.; Prenger, R.; Satheesh, S.; Sengupta, S.; Coates, A.; et al. 2014. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.

Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, 311–318. Association for Computational Linguistics.

Qin, Y.; Carlini, N.; Goodfellow, I.; Cottrell, G.; and Raffel, C. 2019. Imperceptible, robust, and targeted adversarial examples for automatic speech recognition. *arXiv preprint arXiv:1903.10346*.

Shen, J.; Nguyen, P.; Wu, Y.; Chen, Z.; et al. 2019. Lingvo: a modular and scalable framework for sequence-to-sequence modeling.

Smith, S. W. 2019. The scientist and engineer's guide to digital signal processing.

Taori, R.; Kamsetty, A.; Chu, B.; and Vemuri, N. 2019. Targeted adversarial examples for black box audio systems. In *2019 IEEE Security and Privacy Workshops (SPW)*, 15–20. IEEE.