Pony Preservation Project

- Voice -

A Guide

written by **twiggles** !!NPm5BKKNxx4 and many wonderful anons

maintained by **twiggles** !!NPm5BKKNxx4

contribution spreadsheet

current thread

Overview

The Pony Voice Preservation Project is a collaborative effort by /mlp/ to create quality AI text to speech generators for as many My Little Pony: Friendship is Magic characters as possible.

Nine years into the ride and not long before it is scheduled to end, we have reached an age of technology where a trained neural network can make Trump and Obama recite Steamed Hams, or make Jordan Peterson perform Lose Yourself using nothing but recordings of their voices. We have eight (soon to be nine) seasons, five movies, and a leak's worth of voice material available to us in high quality. There is more public research into machine learning than ever before giving us a strong foundation to build upon. By the end of the year the show will have ended leaving the future of the community unclear. Now is the perfect time for this project.

The MLP community has always been extremely creative, making an immense amount of fan content over the span of almost nine years. Having the ability to generate voice lines for a character without needing their voice actor has incredible implications for fan works, and could ensure that new content is still made well beyond the end of the show.

This guide will explain how we plan to accomplish our goal, where we are now, and how you — the person reading this right now — can join in and contribute.

Current Project Goals

- Top Priority: Clip out sections of audio from dialog rips (S1-S8 rips available here)
    - We are currently focusing on Twilight Sparkle's spoken dialogue for our prototype, as she has the most voice material. Feel free to extract full episode audio, but the more we have for a single pony, the more we will be able to experiment with AI early on.
    - Read the contribution instructions then claim an episode in the thread if you want to help out with this.
    - We currently don't have dialog rips for any of the Equestria Girls films. Anyone is free to do this if they want, but please ask about the copy of the movie that you're planning to extract dialog from in the thread first. We are aiming for maximum quality.
- Create a working installation of Tacotron (example implementation on GitHub) or a similar system
    - Various anons have attempted to get Tacotron running on their own systems for training, but many are having trouble with Python dependencies.
        - Several anons in the thread have recommended Jupyter or Conda. Conda comes with many of the packages needed to run Python-based AI systems and provides a solution for dealing with dependencies. Jupyter includes Conda, plus an environment for creating interactive prototypes and demos.
    - There is no organized effort yet. If you have experience with Python and/or machine learning feel free to post ideas and experiment for yourself
- Implement a proper file sharing solution
    - We need some way of archiving and sharing clips, transcriptions, and annotations. A P2P system would be ideal, but unfortunately none of the existing solutions seem to fit our needs (IPFS, BitTorrent, and Syncthing have been looked at to name a few).
    - For the time being, any anons with free hard drive space should download as much as possible to ensure the files won't be lost if the original creator's hard drive fails.
- Research Praat, IPA, X-SAMPA and ToBI
    - In addition to clipping and transcribing dialog we plan to fully annotate the clips with timestamped English and X-SAMPA (or IPA) representations, as well as prosody information with a system such as ToBI. This will give us more power and flexibility in training an AI model.
    - We plan to use a tool called Praat for these annotations. Praat can also be used to extract information about voice pitch from audio files, which could be useful.

Tools

- Audacity, for clipping audio
- Praat, for annotating audio clips with timed transcriptions and ToBI events
- iZotope RX7, for noise removal
    - ISSE is a possible free alternative
    - iZotope RX is not free. However (and I definitely don't endorse this), cracks *do* exist. I won't provide you a link because we want to stay under the radar. Just avoid R2R and CracksNow, they have been known to include ransomware.
- Anon's SRT to Audacity App, for converting SRT subtitle files to Audacity labels
- 14aren, a nice file renaming tool for Windows (for after-the-fact naming adjustments)
- mlp_dialog_rip.sh, the shell script used to extract dialog and subtitles from S01-S08. It applies the process described by this page using the ffmpeg utility
- Anon's Audacity Label Timestamp Script, a PowerShell script to add timestamps to an existing set of Audacity labels
- Anon's Character Tagger, an attempt to automatically tag characters in an Audacity label file by comparing it to transcripts available on the mlp wiki. Due to differences in spelling and abbreviation between transcripts and subtitles, will not be able to tag all lines but should be able to do most.
- Notepad++, a programmer's text editor for Windows, for manually editing Audacity label files
- Anon's Transcript Generator, a Python script for generating transcript text files (instructions available at 33:25 in this demo video)

How to Contribute

Before you start any work, please post about what you plan to do in the thread (e.g. “clipping audio from S01E26”) so other Anons know what’s being worked on.

We are using this spreadsheet to track progress on each episode:
https://derpy.me/EzAMm

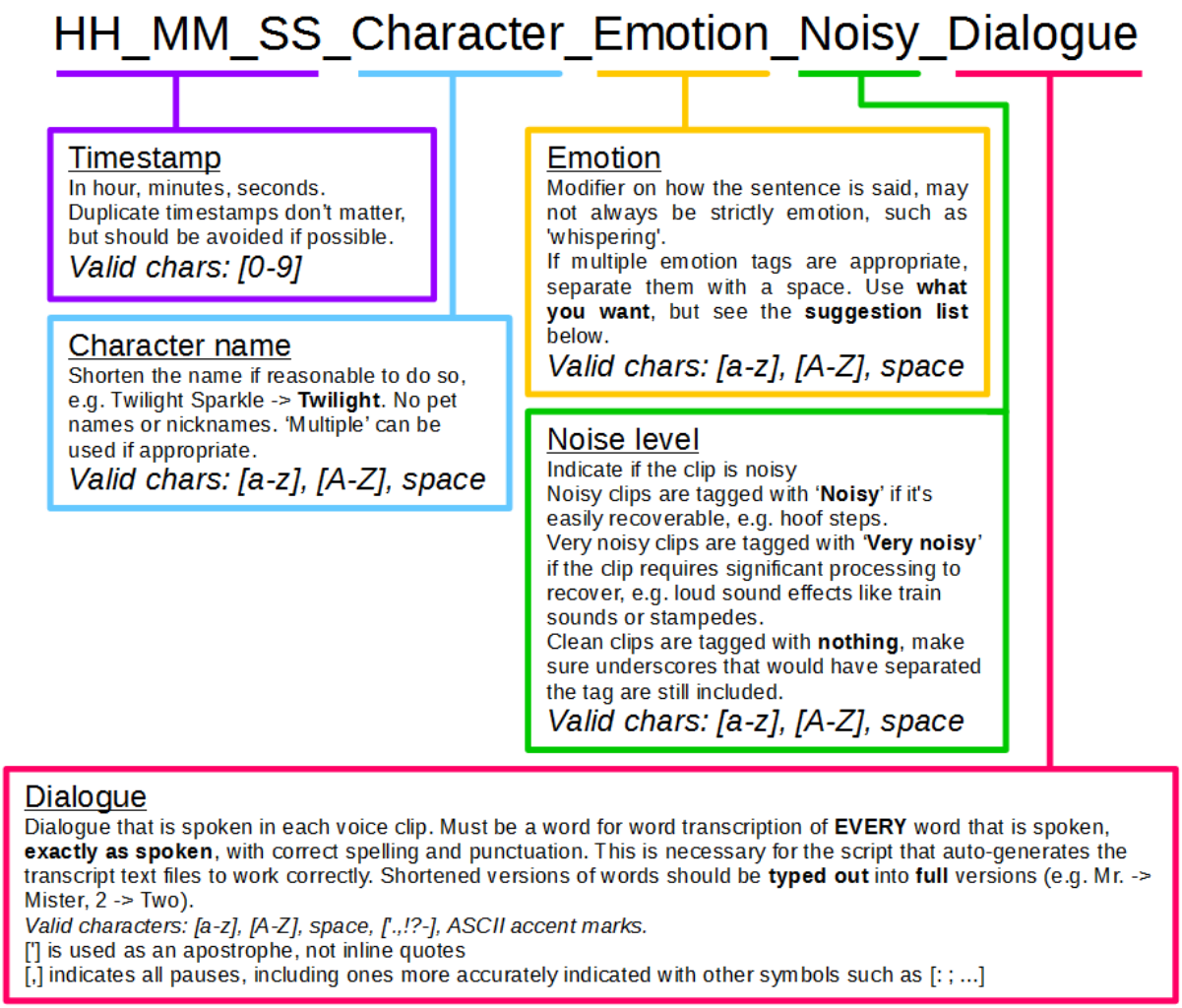
We do not have a place for contributors to upload files at this point, so for the time being simply follow the guidelines below for submitting your content when you are done.

You can submit as much or as little as you want, but audio clips are our top priority. Also keep in mind that we should prioritize episodes with a lot of Twilight Sparkle dialog for the time being, as the sooner we have enough data to train a model the sooner we can create a working proof of concept.

File Name Conventions

Phase 2 is using the high quality audio track from the episodes and films to extract all the chucks of dialogue and speech and link it to its exact transcript. For each episode / film, the end product is a shitload of audio flac files (the dialogue / speech itself) and an audacity label file (which contains all the exact transcript and very precise start and end time of said sample in the original audio).
By using a precise notation, each audio chunk can then be used for machine learning (or everything else if you will).

TI ; Dr : Name your files like this fags :



>Everything is case insensitive, upper case is used for human readability only.
>Global allowed characters are the following ASCII: [a-z], [A-Z], [0-9], [ ], [_-,.!?'], accent marks.
>Disallowed characters: "#$%&()*+/:;<+>@[]\^{}|~, ASCII control characters, DEL, All extended ASCII codes except accent marks. Underscores only for separation (and question mark, see below)
>[?] is replaced in file names with _, this is done automatically by Audacity in the export phase. This is unambiguous as we know that every underscore after the sixth one should be a question mark.
>The text file containing the raw Audacity labels must be included in all submissions as '**labels.txt**'.
>Each episode will have its clips in its own '**SXXEXX**' folder, so we know where each line came from. This will allow easy addition of extra identifiers in file names later by script if desired.

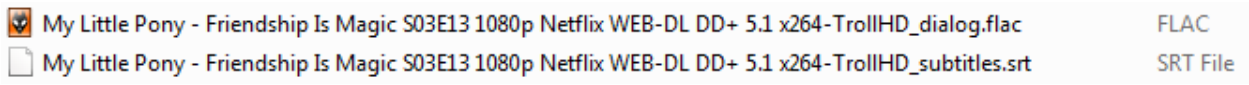
Unfortunately Windows, being a shit OS, has a hard file name limit of 260 characters. Your description should be long enough to summarize the clip's contents, but not so long that the full file name exceeds this limit. This is to ensure that as many anons as possible can contribute.

Store your files in folders labeled with the source of the clips (season and episode if from the show, name of the alternate source otherwise).

Clipping Audio

Our tool of choice for clipping audio is Audacity, a free and open source audio editor. It allows you to create time labels for sections of audio and export those sections to separate files all at once. One generous anon even wrote a tool that can generate Audacity labels from the SRT subtitle files in the dialog rips (link in the Tools section), which should give you a considerable head start with labeling audio clips.

The only high quality material available to us at the time of writing is the S1-S8 dialog torrent linked in the resources section. This torrent contains both a FLAC audio file and an SRT subtitle file for each episode of each season. We will continue this demonstration using a pair of files from that torrent as an example, say the infamous season 3 episode 13:



Anons have pointed out that there are issues with the subtitles in some of the episodes of the S1-S8 rips (mainly dialog being repeated). If an episode you are working on has this problem, there is an alternate subtitle source in the Resources section.

Our first step will be to generate labels for the dialog in our episode.

Using the SRT to Audacity Tool

Upon loading Anon's SRT to Audacity tool you will be greeted with this interface.

Begin by clicking "Choose File" and selecting the .srt file for your chosen episode. This will automatically copy the contents into the text box. The three checkboxes in the bottom left corner will be unchecked at first, but I highly recommend that you check all of them as in the screenshot. This will make your job much easier. The prefix and suffix text boxes can be used to add additional text to each label, such as a character name, but since we're processing an entire SRT file we will not add any.

If you forgot to check the "Label Timestamp" option before exporting, you can use Anon's Audacity Label Timestamp Script in the Tools section.

Several anons have mentioned that there are far more noisy clips than clean ones, and that adding "noisy" to the end of almost every label in Audacity is time consuming. If you find this to

be the case, I recommend adding “noisy” as a suffix to all labels and simply removing it when it’s not needed.

Click “Start” to convert the subtitles to the Audacity label format, then “Save” to save the results to a file. Give the file a useful name and save it with a .txt extension.

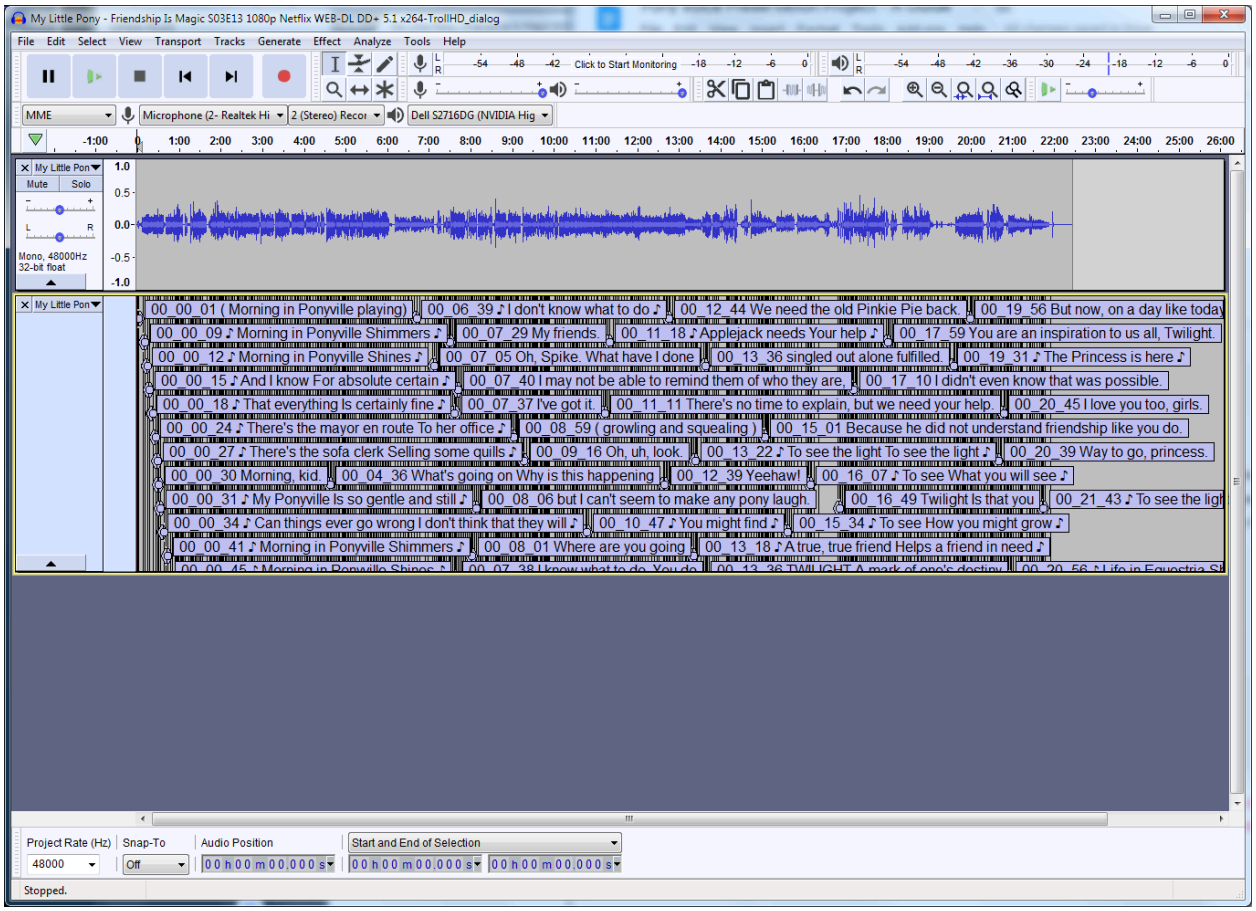
Creating Clips in Audacity

Demonstration Video: https://www.youtube.com/watch?v=NR0ks0pVb9k

Narrated Tutorial: https://youtu.be/4vuQ0cXK9fI

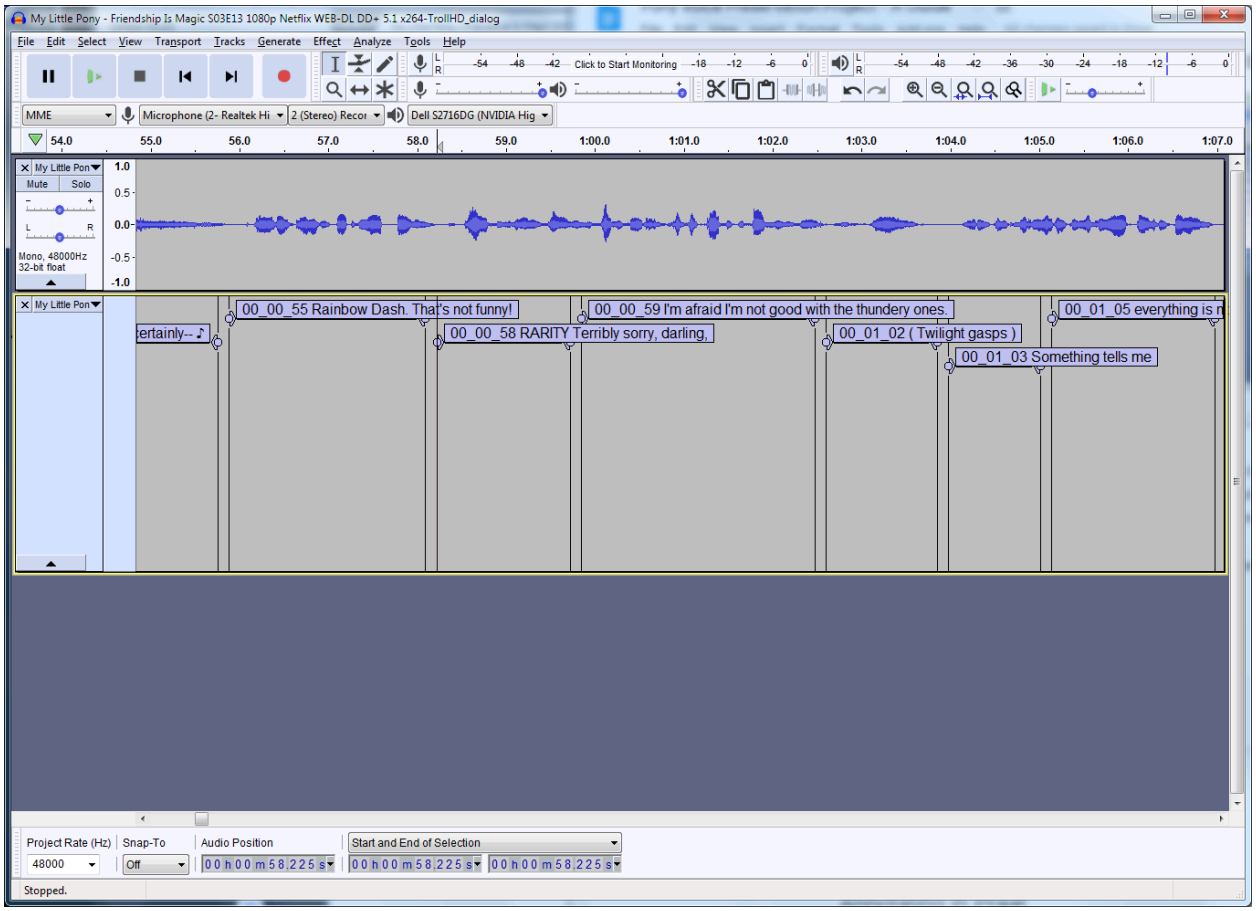
Video Tutorial Series (w/ transcription script): youtube.com/watch?v=KmpXyBbOObM

Open the episode dialog audio file in Audacity. Once the audio has finished loading select “File” ->” Import” -> ”Labels” in the menu. Choose the label file that we just created in the previous step. Your Audacity window should now look something like this, with one track representing the audio and another representing the labels:



By default the zoom level is set so that you can see the entire time range at once. This is not useful, as episodes are 22 minutes long. You can zoom in and out to get a closer look using the magnifying glass icons or by holding Ctrl and scrolling with your cursor over either track.

Upon zooming in (plus skipping and deleting the beginning of the episode, as it starts with a song) we can much more clearly see the individual labels:



The black vertical bars in the clip track represent label boundaries, beginning and end. Clicking and dragging on either will allow you to start a selection from that point. The circular buttons on either end of the label text allow you to move the entire label in time, and the triangles attached to the circles allow you to move the beginning or end individually.

At any point you can press the spacebar to begin audio playback from wherever you last clicked on the timeline. If you select a region and then press space only the selected region will be played back. Use this to adjust the label timing as needed, the subtitles are not perfect. You may combine labels as needed if the audio doesn't split well on the existing bounds between two labels, though there's no automatic way of doing this. Simply right click on and delete the second label and extend the first one to compensate.

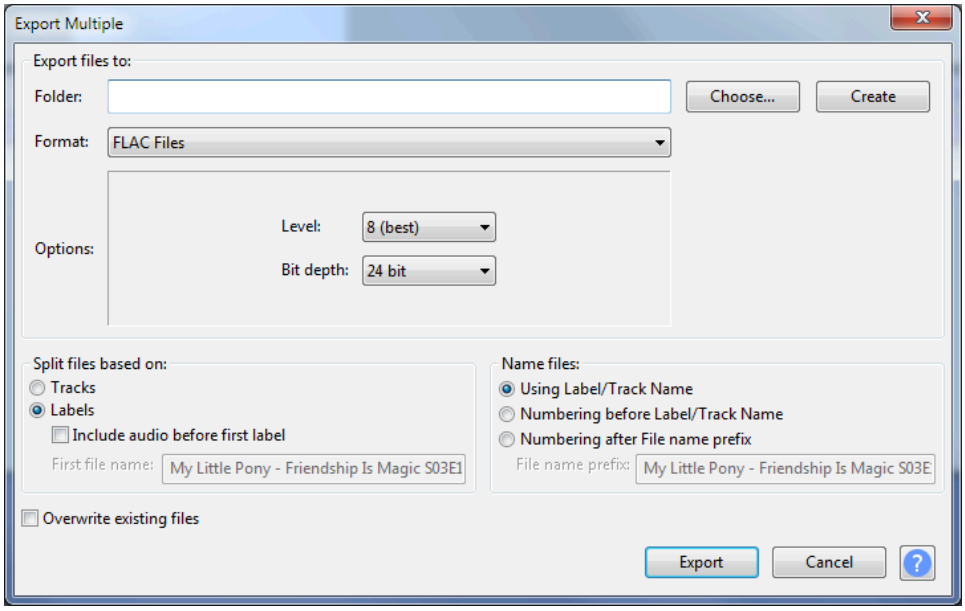
Audio clips should contain about a sentence worth of dialogue, but it is most important that your clips are split in a way that sounds natural. Do not let clips start part way into a word or let them end in the middle of one. Try to clip such that the emotion conveyed by the tone of voice is consistent throughout. Audio clips should also contain entirely noisy or entirely clean audio, so take this into account as well when deciding where to put your clip boundaries.

Follow the file naming conventions above while entering label text names. You can use this key for abbreviating character names and emotions: https://pastebin.com/3ev1Q0xD

As the anon who wrote the key pointed out, not all characters are included and there may be some cases where the emotion tags given don't quite fit. Use your best judgement.

If any of the label names become exceptionally long, you may remove words from the end of the dialogue to shorten them. Don't forget to adjust the timestamp of the label if you shift its beginning significantly.

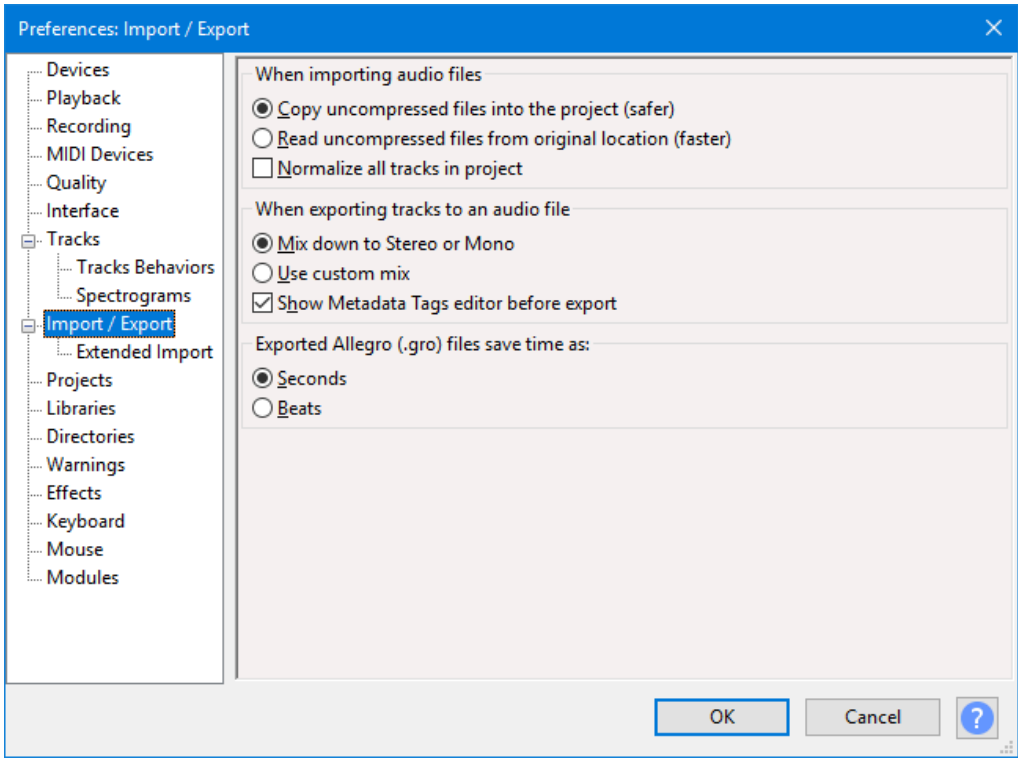
Finally, select "File" -> "Export" -> "Export Multiple" to begin the process of exporting audio clips.



Set your export options to match those in the above screenshot. We have decided to use the FLAC audio format to ensure that audio quality is maintained and files remain as small as

possible in the process. I recommend setting your export destination to a new folder titled with the season number and the episode number (or the name of the source in the case of a movie).

I also highly recommend disabling “Show Metadata Tags editor before export” in the “Edit” -> “Preferences” -> “Import Export” menu to prevent a window popping up for every single label in your project.



Click “Export” in the Export Multiple window when you are finished. You should also export and submit your newly changed labels under “File” -> “Export” -> “Export Labels”.

Transcribing Clips

This is probably the easiest (but still time consuming) step of clipping. For each clip that you exported from Audacity, create a plain text file containing a full English transcription of the dialog in the clip. You may use the file name and/or subtitles as a base, but you must edit as necessary.

Use proper punctuation and expand abbreviations to their full forms (e.g. “Mister” instead of “Mr.”).

We highly recommend that you use Anon’s Transcription Generator in the Tools section in the tools section to help perform this step automatically. Video instructions are available at the beginning of the Creating Clips in Audacity section.

Cleaning up Audio

To be written… (iZotope RX has official tutorials available if you want to try this.
An audio anon has recommended trying the “Music Rebalance” option with these settings to quickly reduce the volume of non-vocal elements, though you should check if this makes the audio sound unnatural. RX is also capable of precise manual noise removal when necessary)

Annotating in Praat

To be written… (this step is not necessary yet, just audio clips and text transcription for now)

Submitting Your Content

We do not have an official solution for sharing contributions at this point. For the time being, put your files (audio, text transcriptions, modified Audacity labels, etc) in a Zip archive, upload it, and post a link in the thread. Explain what it is you're uploading, and make sure that the files are sorted into folders by season and episode (or by source if not from the show).

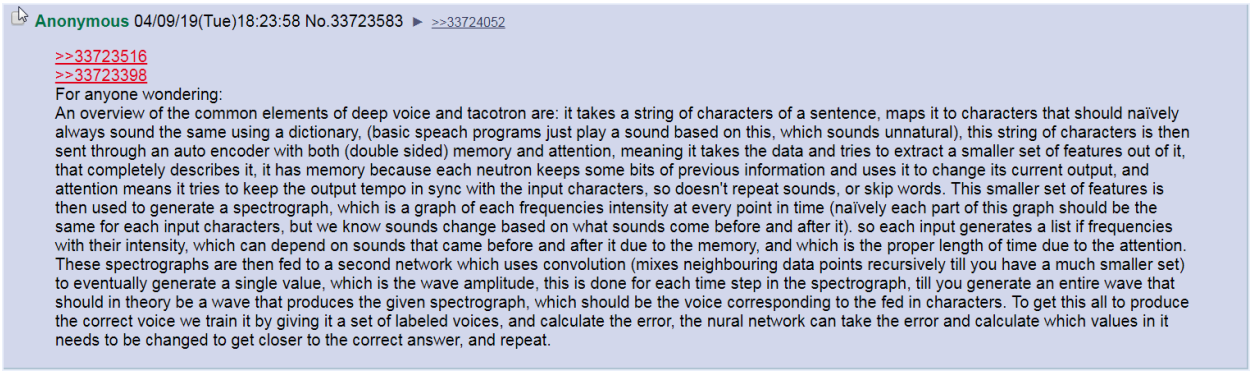
You may also leave a comment on the contribution spreadsheet.

If you're uploading cleaned audio from a noisy clip that you made, upload both versions of the clip. We will be archiving both the original and a cleaned version so that anons can create their own cleaned versions if they feel they can do better.

Some suggested file hosts:
https://smutty.horse
https://zippyshare.com
https://nofile.io
https://anonfile.com
https://mega.co.nz
https://dropbox.com
https://drive.google.com

Our Plan (WIP)

Machine learning works as the name implies: you create a program (often a neural network) that can learn from information you give it and “teach” it what outputs it should return for a given input. There are already a large number of open source machine learning projects available to use as a foundation, including several specifically designed for what we are trying to do (e.g. Google’s Tacotron). This Anon explains it in more detail:



Anonymous 04/09/19(Tue)18:23:58 No.33723583 ► >>33724052

>>33723516
>>33723398
For anyone wondering:
An overview of the common elements of deep voice and tacotron are: it takes a string of characters of a sentence, maps it to characters that should naïvely always sound the same using a dictionary, (basic speach programs just play a sound based on this, which sounds unnatural), this string of characters is then sent through an auto encoder with both (double sided) memory and attention, meaning it takes the data and tries to extract a smaller set of features out of it, that completely describes it, it has memory because each neutron keeps some bits of previous information and uses it to change its current output, and attention means it tries to keep the output tempo in sync with the input characters, so doesn't repeat sounds, or skip words. This smaller set of features is then used to generate a spectrograph, which is a graph of each frequencies intensity at every point in time (naïvely each part of this graph should be the same for each input characters, but we know sounds change based on what sounds come before and after it). so each input generates a list if frequencies with their intensity, which can depend on sounds that came before and after it due to the memory, and which is the proper length of time due to the attention. These spectrographs are then fed to a second network which uses convolution (mixes neighbouring data points recursively till you have a much smaller set) to eventually generate a single value, which is the wave amplitude, this is done for each time step in the spectrograph, till you generate an entire wave that should in theory be a wave that produces the given spectrograph, which should be the voice corresponding to the fed in characters. To get this all to produce the correct voice we train it by giving it a set of labeled voices, and calculate the error, the nural network can take the error and calculate which values in it needs to be changed to get closer to the correct answer, and repeat.

A project such as Tacotron could be used to produce a proof of concept, but first we need a LOT of audio to work with. Dialog has already been extracted from MLP:FIM seasons one through eight in the highest quality available. There is a torrent available for this in the “Resources” section of this guide.

Resources

IF YOU USE ANY OF THE TORRENT FILES PLEASE LEAVE THEM SEEDING FOR OTHER ANONS

MLP:FIM Ripped Dialog S1-S8:

https://files.catbox.moe/2q4p2z.torrent

Alternate subtitle source if things don't seem right (Stange doubling):

https://www.addic7ed.com/

Full Episodes with E-AC3 640kbps audio from Amazon rips for seasons 1-2:

http://mlpeps.com/mlp/testfiles/amzn%20webrips/

Full Episodes with E-AC3 640kbps audio from Netflix rips for seasons 3, 5, 6, 8:

https://rarbg2018.org/torrent/7iz2nyw

https://rarbg2018.org/torrent/e2kvr3f

https://rarbg2018.org/torrent/hj63unr

https://rarbg2018.org/torrent/utoqs46

Movie Rips:

Blu-Ray x264 Replica https://www.filecrypt.cc/Container/3BF6F363FB.html (filecrypt/zippyshare)

DTS@768kbps 6 channel

Special Source ^:)

https://anonfile.com/Sbveyfk3n4/12_angry_chinese_men_yell_at_each_other_for_13_hours_7z

6912kbps 24bit FLAC 6 Channel

~~Password: (You guys are an insightful bunch, I'm sure you can figure it out)~~

(Old) Project Logo Photoshop File:

https://files.catbox.moe/j50wq1.psd

Anon's In-Depth Mane 6 Vocal Profiles:

https://drive.google.com/drive/folders/12wXl80FDHpJTXSE5Sw5fF_6Fm0GxtC6V?usp=sharing

EQG Audio: https://anonfile.com/M6s4u8sfne/EGG_7z

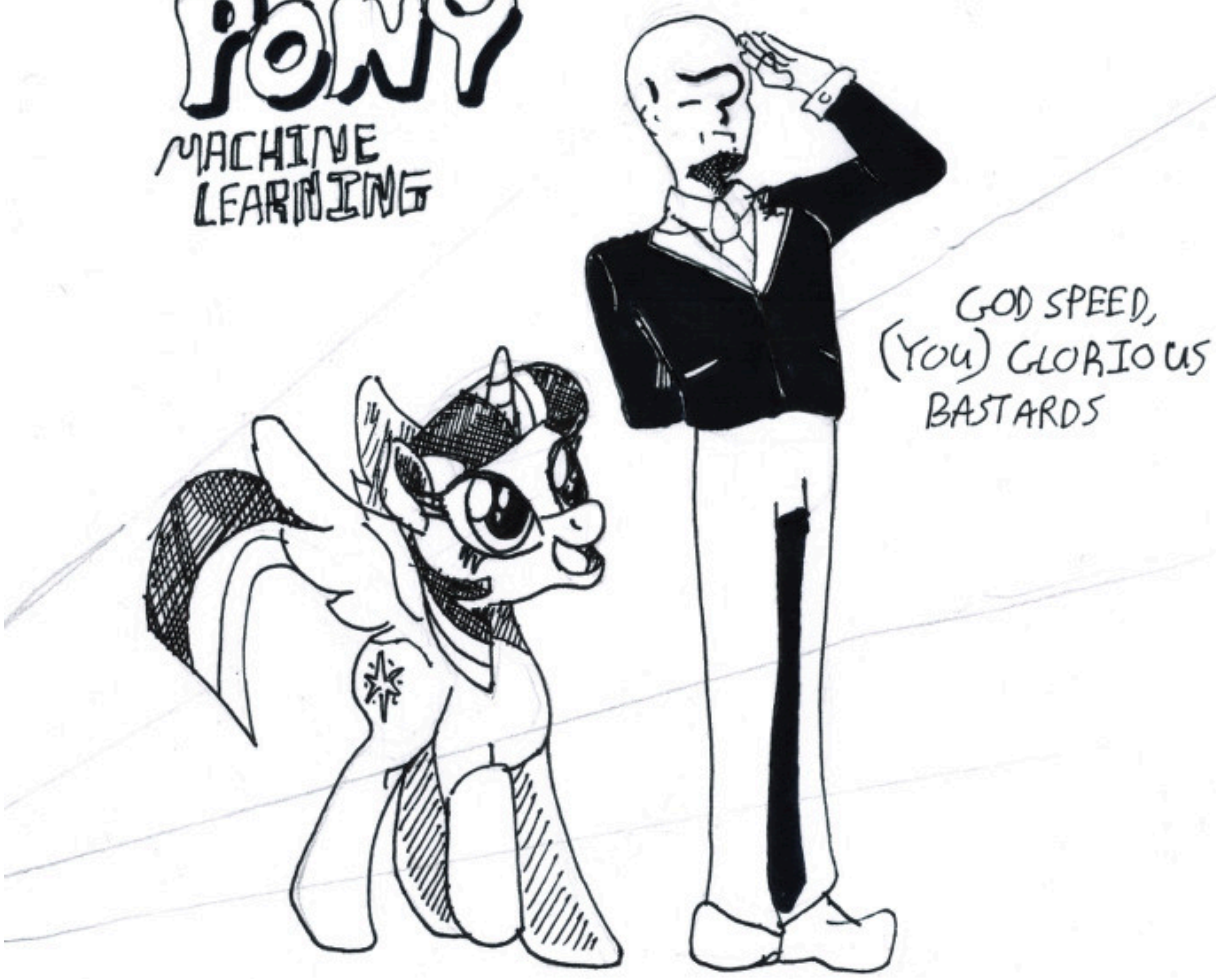
Changelog

- 05/16/19: Updated clipping instructions
- 05/16/19: Updated video demonstration link in clipping instructions
- 05/16/19: Replaced file naming conventions with updated image from thread
- 05/16/19: Updated link to Anon's Transcript Generator in Tools
- 04/27/19: Added new series of video tutorials to Clipping Audio
- 04/27/19: Added Anon's transcript generator to Tools
- 04/20/19: Implemented various suggested changes to clean up the guide somewhat
- 04/18/19: Added Anon's narrated video demo to Clipping Audio
- 04/16/19: Added new tracker to S1-S8 dialog torrent file, replaced old link
- 04/15/19: Added links to contribution spreadsheet
- 04/15/19: Added Anon's video demo to Clipping Audio
- 04/15/19: Added list of episodes being worked on/complete to How to Contribute
- 04/14/19: Added logo Photoshop file to resources
- 04/14/19: Added officially endorsed file naming scheme
- 04/14/19: Added alternate subtitle source to resources
- 04/12/19: Added clarification about what counts as background noise for marking clips
- 04/12/19: Added Audio Anon's super special lossless movie source to Resources
- 04/12/19: Added Anon's PowerShell script for adding timestamps to Audacity labels
- 04/12/19: Added ffmpeg-based dialog extractor script to Tools
- 04/12/19: Added changelog section
- 04/12/19: Added tip to audio cleanup section for quickly removing noise in RX
- 04/12/19: Clarified how clip beginnings and ends should be decided, and how long clips should be
- 04/12/19: Added tip for easily marking files as noisy, in the case that the majority are

/MLP/ 04/12/19 9:30pm CT

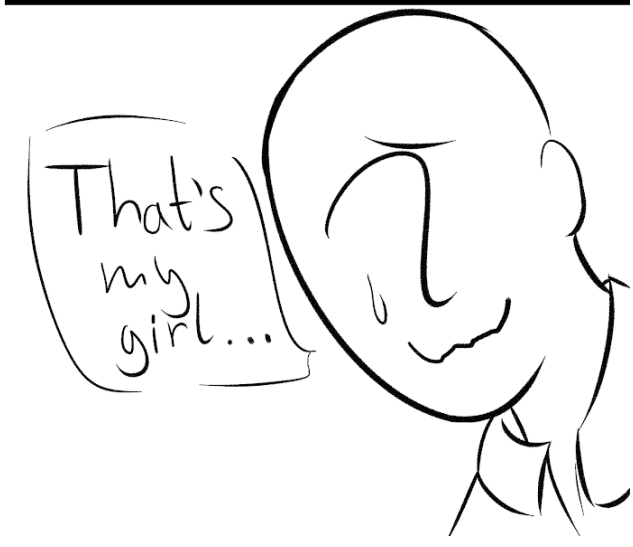
PONY

MACHINE LEARNING



Test Run Audio Demos: https://files.catbox.moe/91f1w1.zip

Come on, TwAIlight, say "cake"


That's my girl...

https://clyp.it/mmxgq1yx