

CHAPTER 10: SPEECH SYNTHESIS

INTRODUCTION TO SPEECH SYNTHESIS

Speech synthesis can be used to add additional lines of dialogue to NPCs that use base game voice types.

In Wyrmsooth, I use speech synthesis to give background NPCs, such as the hunters at Hunter's Shack or the marauder leader at Cragwater Camp, unique things to say.

In this section I'll be showing you how to train a custom Tacotron and WaveGlow model on the Google Colab platform using a dataset based on a voice type from The Elder Scrolls V: Skyrim.

Tacotron is a generative text-to-speech synthesis program. For this tutorial I'll be using the NVIDIA Tacotron 2 repository available on GitHub: <https://github.com/NVIDIA/tacotron2>.

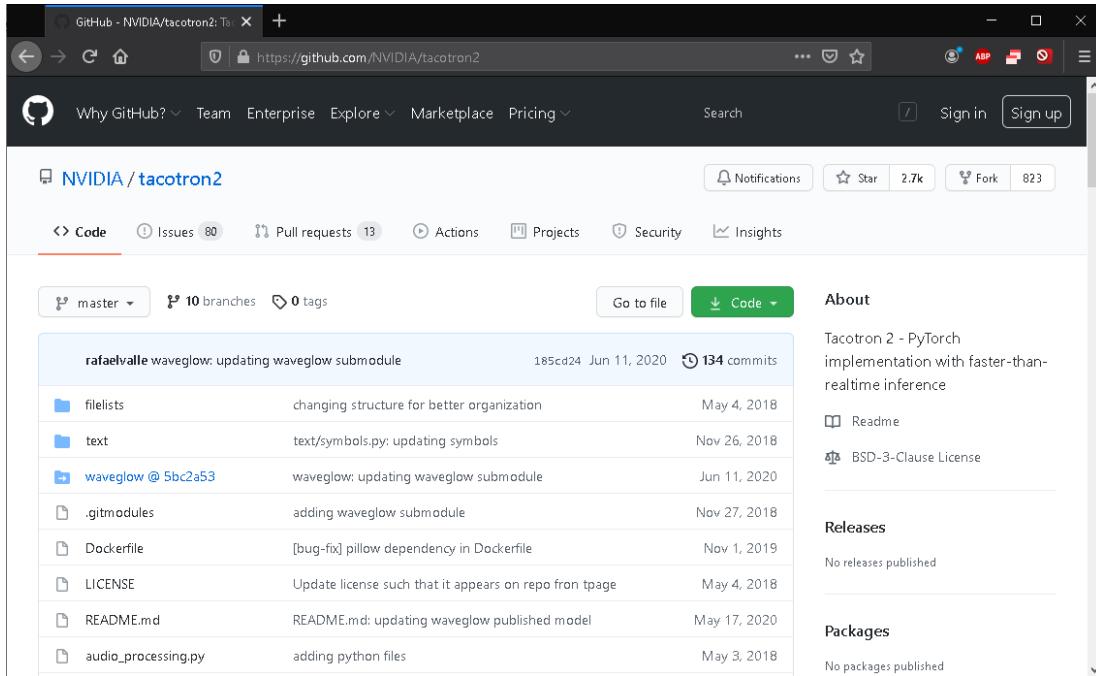


Figure 1308 - NVIDIA Tacotron 2 on GitHub.

WaveGlow is a flow-based generative speech-synthesis program. A custom WaveGlow model will significantly improve the quality of synthesized speech. For this tutorial I'll be using the NVIDIA WaveGlow repository available on GitHub: <https://github.com/NVIDIA/waveglow>.

Google Colab is a cloud platform that I'll be using in this tutorial to run Tacotron and WaveGlow. It can be used to run python code and other commands in a pre-built python environment via your browser. We can link our Google Drive storage to a Colab session so we can access our dataset and save checkpoints back to it as we train.

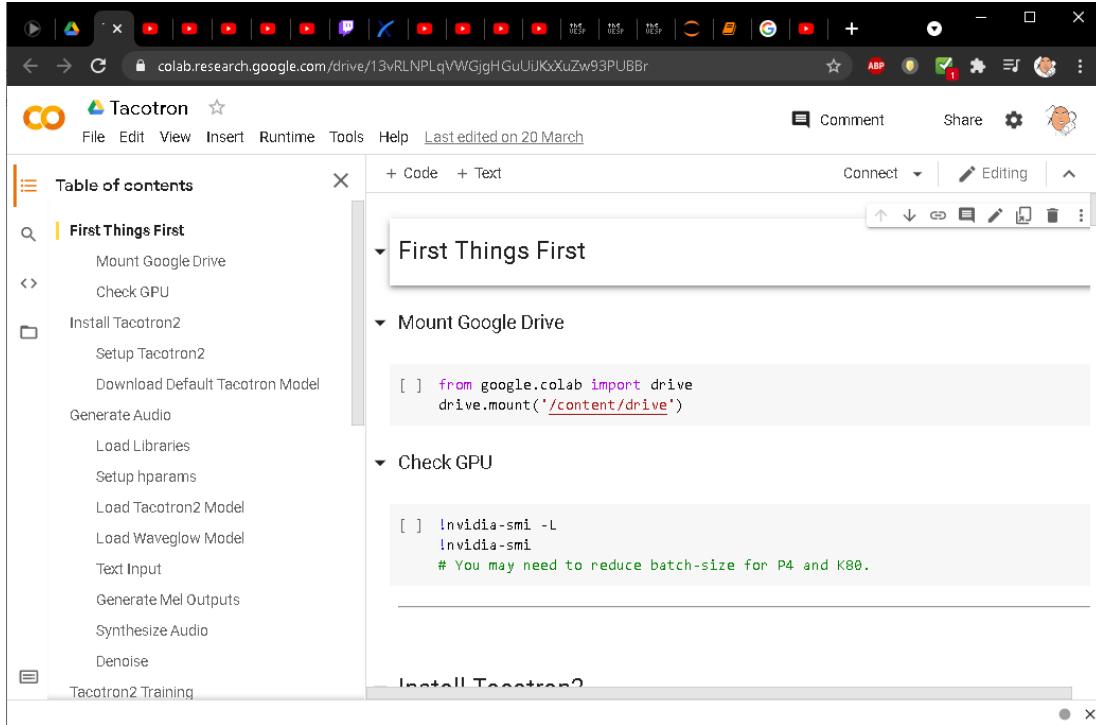


Figure 1309 - A Colab Notebook.

I've already set up a Colab Notebook with all the commands you'll need to run. You can access it here:

<https://colab.research.google.com/drive/13vRLNPLqVWGjgHGuUiJKxXuZw93PUBBr?usp=sharing>

The main advantage of using Google Colab is that we don't need to set up our own python environment locally.

Here is some sample output from a few different models I've trained:

[tacotron_sample_femalenord1.wav](#)
[tacotron_sample_femalenord2.wav](#)
[tacotron_sample_femalenord3.wav](#)
[tacotron_sample_joerogan1.wav](#)
[tacotron_sample_joerogan2.wav](#)
[tacotron_sample_joerogan3.wav](#)
[tacotron_sample_femalecommander1.wav](#)
[tacotron_sample_femalecommander2.wav](#)
[tacotron_sample_femalecommander3.wav](#)
[tacotron_sample_malecommander1.wav](#)
[tacotron_sample_malecommander2.wav](#)
[tacotron_sample_malecommander3.wav](#)
[tacotron_sample_maleslycynical1.wav](#)
[tacotron_sample_maleslycynical2.wav](#)
[tacotron_sample_maleslycynical3.wav](#)

As you can hear, Tacotron is able to pick up on various vocal nuances like the rolling r's of the femalenord voice actor's accent.

PREPARING A DATASET USING VOICE ACTING FROM THE ELDER SCROLLS V: SKYRIM

In this section I'll be showing you how to prepare a new dataset from which we'll be training a new Tacotron and WaveGlow model.

A dataset consists of voice clips from a single speaker and their corresponding subtitles. Generally speaking, the more voice acting we have the better our results should be. Models based on datasets consisting of less than 3 hours of audio are going to struggle with articulation.

The first thing we need to do is extract the subtitle files from The Elder Scrolls V: Skyrim.

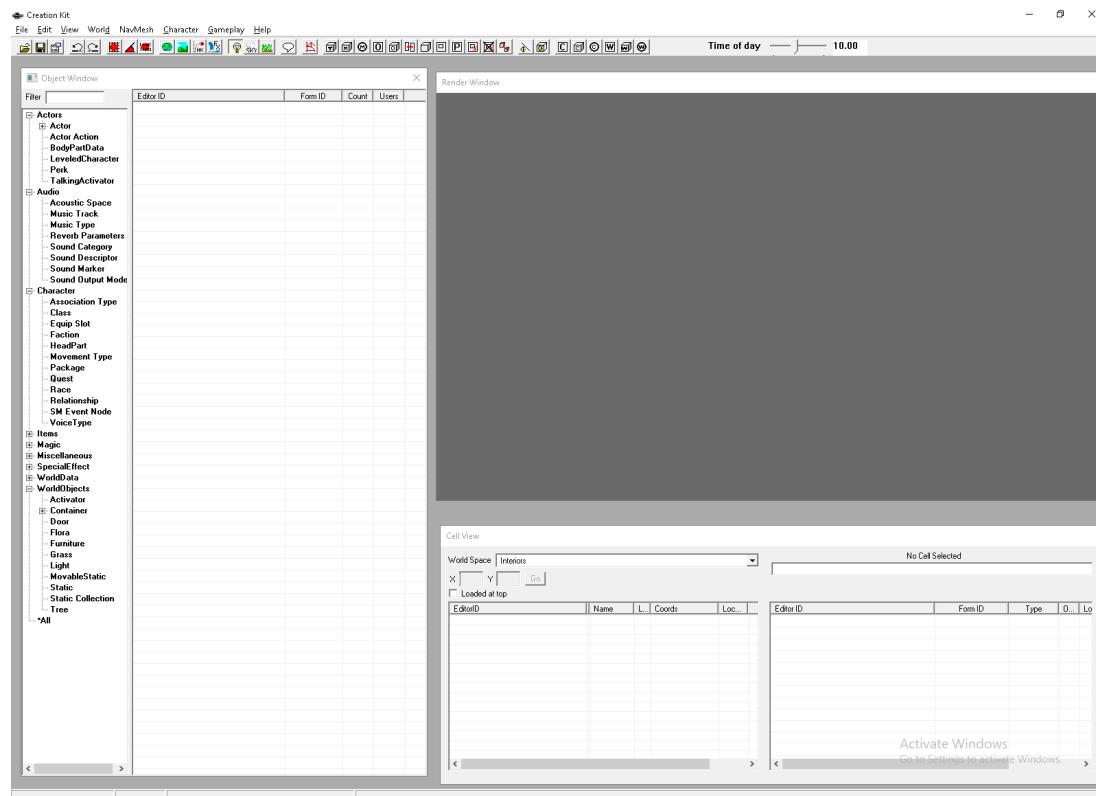


Figure 1310 - Creation Kit.

Once the Creation Kit loads, go to File > Data. Double-click on ‘Skyrim.esm’ then click OK and wait for it to load.

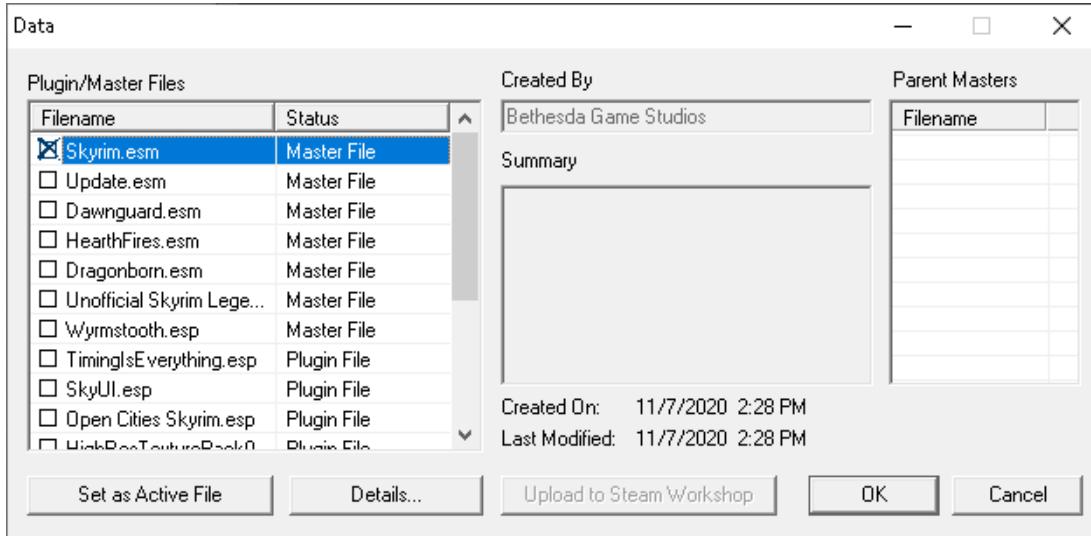


Figure 1311 - Loading Skyrim.esm.

Click ‘Yes to All’ to any warnings that pop up.

Go to Character > Export Dialogue and click OK to the Export Dialogue popup window.

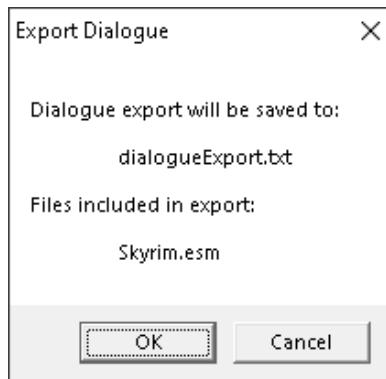


Figure 1312 - Exporting dialogue.

This will create a dialogueExport.txt file in your Skyrim installation folder.

Again, click ‘Yes to All’ to any warnings that pop up. Close the Creation Kit once the export has finished.

For this next part we’ll need **Notepad++** and **Microsoft Excel**, or a similar spreadsheet application. You can download Notepad++ for free from its website: <https://notepad-plus-plus.org/downloads/>.

Open dialogueExport.txt in Notepad++. It should look like this:

```

C:\Program Files (x86)\Steam\steamapps\common\Skyrim\dialogueExport.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
dialogueExport.txt

1 LINE ID SPEAKER NPCID GetIsID RACE VOICE TYPE QUEST BRANCH CATEGORY TYPE SUBTYPE TOPIC TO
2 FFFFFFFF -- -- -- -- CrWerewolfVoice CreatureDialogueWerewolf (none) Combat Combat Death
3 FFFFFFFF -- -- -- -- CrWerewolfVoice CreatureDialogueWerewolf (none) Combat Combat Hit Fo
4 FFFFFFFF -- -- -- -- CrFoxVoice CreatureDialogueFox (none) Combat Combat Death FormID 001
5 FFFFFFFF -- -- -- -- CrFoxVoice CreatureDialogueFox (none) Combat Combat Hit FormID 0010F66
6 FFFFFFFF -- -- -- -- CrChickenVoice CreatureDialogueChicken (none) Combat Combat Death Fo
7 FFFFFFFF -- -- -- -- CrChickenVoice CreatureDialogueChicken (none) Combat Combat Hit FormID
8 FFFFFFFF -- -- -- -- CrChickenVoice CreatureDialogueChicken (none) Miscellaneous Miscellaneous
9 FFFFFFFF -- -- -- -- FemaleUniqueDelphine MQSkyHavenSparring (none) Miscellaneous Miscellaneous
10 FFFFFFFF -- -- -- -- MaleOldKindly DialogueWinterholdCollegePostSceneQuest (none) Scene Scened
11 FFFFFFFF -- -- -- -- FemaleSultry DialogueWinterholdCollegePostSceneQuest (none) Scene Scened
12 FFFFFFFF -- -- -- -- MaleEvenToned DialogueWinterholdCollegePostSceneQuest (none) Scene Scened
13 FFFFFFFF -- -- -- -- MaleCondescending DialogueWinterholdCollegePostSceneQuest (none) Scene Sc
14 FFFFFFFF -- -- -- -- MaleYoungEager DialogueWinterholdCollegePostSceneQuest (none) Scene Scened
15 FFFFFFFF -- -- -- -- FemaleYoungEager DialogueWinterholdCollegePostSceneQuest (none) Scene Sc
16 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 WERoad11Branch02 Topic PlayerDialogue
17 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 WERoad11Branch04 Topic PlayerDialogue
18 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 WERoad11Branch01 Topic PlayerDialogue
19 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 WERoad11Branch01 Topic PlayerDialogue
20 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 WERoad11Branch01 Topic PlayerDialogue
21 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 WERoad11Branch03 Topic PlayerDialogue
22 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 WERoad11Branch03 Topic PlayerDialogue
23 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 (none) Miscellaneous Miscellaneous He
24 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 (none) Miscellaneous Miscellaneous He
25 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 (none) Miscellaneous Miscellaneous Sh
26 FFFFFFFF -- -- -- -- MaleEvenTonedAccented WERoad11 (none) Miscellaneous Miscellaneous Sh
27 FFFFFFFF -- -- -- -- MaleArgonian WERoad09 WERoad09Offer Topic PlayerDialogue Custom WE

```

Normal text file length : 20,770,570 lines : 59,608 Ln:1 Col:1 Pos:1 Unix (LF) UTF-8 INS

Figure 1313 - dialogueExport.txt.

We'll need to remove unnecessary columns first. Go to Search > Replace. In the 'Find what' field enter a comma ',' and in the 'Replace with' field enter an at symbol '@' then click Replace All.

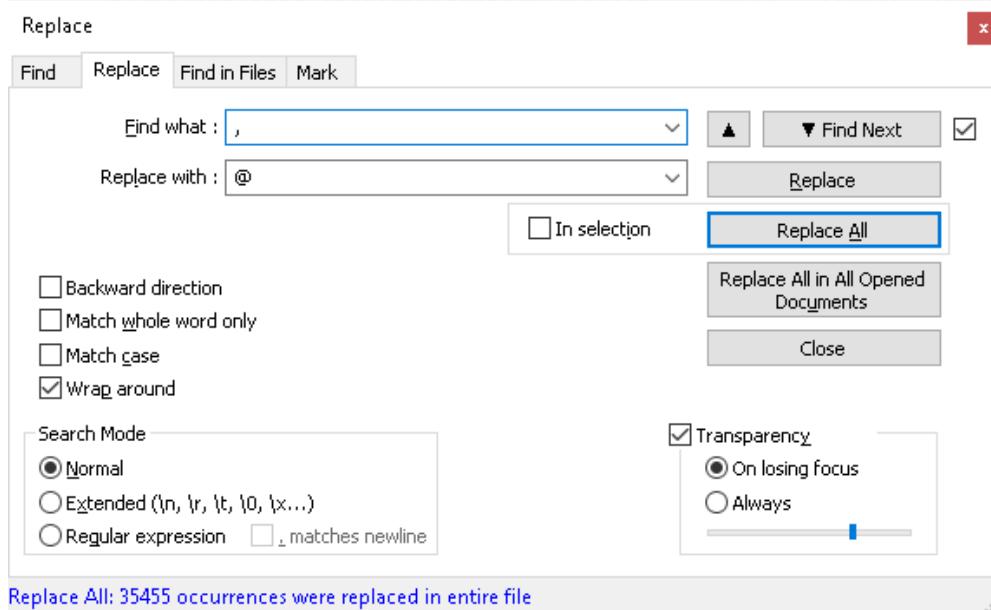


Figure 1314 - Replacing commas with @ symbols.

Don't worry, we'll be reversing that change once we're done in Excel.

Keep Replace open. This time, in the 'Find what' field enter a tab character ' ` ', and in the 'Replace with' field enter a comma ',' then click Replace All.

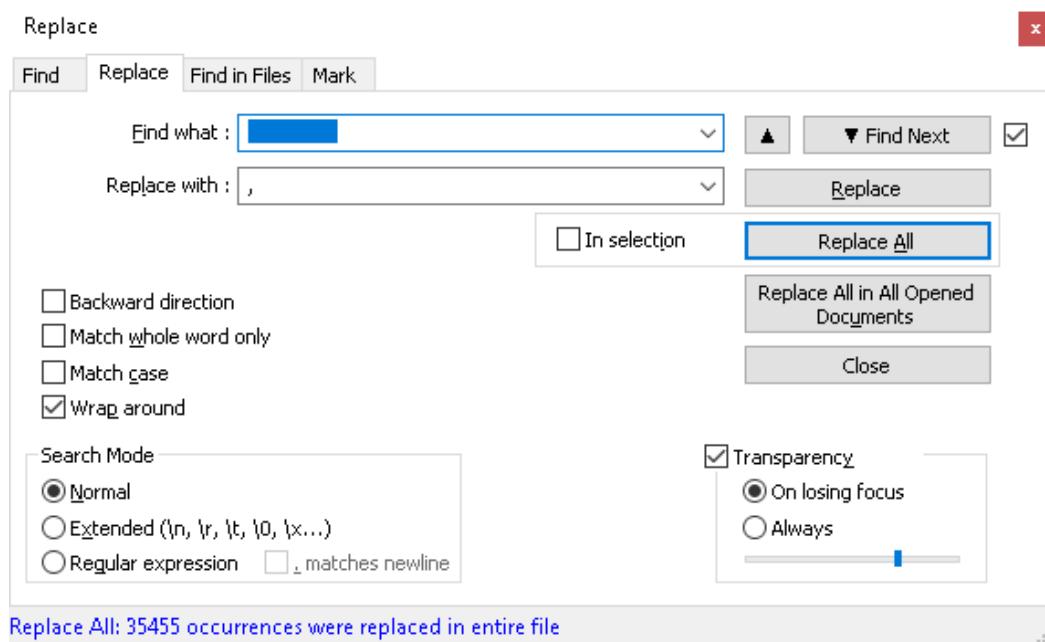


Figure 1315 - Replacing tabs with commas.

Our dialogueExport.txt file should now look like this:

```
*C:\Program Files (x86)\Steam\steamapps\common\Skyrim\dialogueExport.txt - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
dialogueExport.txt

1 LINE_ID, SPEAKER, NPCID, GetIsID, RACE, VOICE_TYPE, QUEST, BRANCH, CATEGORY, TYPE, SUBTYPE, TOPIC, TOPICINFO, RESPONSE
2 FFFFFFFF, --, --, --, --, CrWerewolfVoice, CreatureDialogueWerewolf, (none), Combat, Combat, Death,, FormID 00000E44,
3 FFFFFFFF, --, --, --, --, CrWerewolfVoice, CreatureDialogueWerewolf, (none), Combat, Combat, Hit,, FormID 00000E45,1,
4 FFFFFFFF, --, --, --, --, CrFoxVoice, CreatureDialogueFox, (none), Combat, Combat, Death,, FormID 0010F670,1,Creature
5 FFFFFFFF, --, --, --, --, CrFoxVoice, CreatureDialogueFox, (none), Combat, Combat, Hit,, FormID 0010F66F,1,CreatureDi
6 FFFFFFFF, --, --, --, --, CrChickenVoice, CreatureDialogueChicken, (none), Combat, Combat, Death,, FormID 0010AA5C,1,
7 FFFFFFFF, --, --, --, --, CrChickenVoice, CreatureDialogueChicken, (none), Combat, Combat, Hit,, FormID 0010AA5B,1,Cr
8 FFFFFFFF, --, --, --, --, CrChickenVoice, CreatureDialogueChicken, (none), Miscellaneous, Miscellaneous, Idle,, FormI
9 FFFFFFFF, --, --, --, --, FemaleUniqueDelphine, MGSkyHavenSparring, (none), Miscellaneous, Miscellaneous, SharedInfo
10 FFFFFFFF, --, --, --, --, MaleOldKindly, DialogueWinterholdCollegePostSceneQuest, (none), Scene, SceneDialogue, Cust
11 FFFFFFFF, --, --, --, --, FemaleSultry, DialogueWinterholdCollegePostSceneQuest, (none), Scene, SceneDialogue, Custo
12 FFFFFFFF, --, --, --, --, MaleEvenToned, DialogueWinterholdCollegePostSceneQuest, (none), Scene, SceneDialogue, Cust
13 FFFFFFFF, --, --, --, --, MaleCondescending, DialogueWinterholdCollegePostSceneQuest, (none), Scene, SceneDialogue,
14 FFFFFFFF, --, --, --, --, MaleYoungEager, DialogueWinterholdCollegePostSceneQuest, (none), Scene, SceneDialogue, Cus
15 FFFFFFFF, --, --, --, --, FemaleYoungEager, DialogueWinterholdCollegePostSceneQuest, (none), Scene, SceneDialogue, C
16 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, WERoad11Branch02, Topic, PlayerDialogue, Custom, WERoad11B
17 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, WERoad11Branch04, Topic, PlayerDialogue, Custom, WERoad11B
18 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, WERoad11Branch01, Topic, PlayerDialogue, Custom, WERoad11B
19 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, WERoad11Branch01, Topic, PlayerDialogue, Custom, WERoad11T
20 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, WERoad11Branch01, Topic, PlayerDialogue, Custom, WERoad11T
21 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, WERoad11Branch03, Topic, PlayerDialogue, Custom, WERoad11B
22 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, WERoad11Branch03, Topic, PlayerDialogue, Custom, WERoad11B
23 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, (none), Miscellaneous, Miscellaneous, Hello, WERoad11Hello
24 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, (none), Miscellaneous, Miscellaneous, Hello, WERoad11Hello
25 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, (none), Miscellaneous, Miscellaneous, SharedInfo, WERoad11
26 FFFFFFFF, --, --, --, --, MaleEvenTonedAccented, WERoad11, (none), Miscellaneous, Miscellaneous, SharedInfo, WERoad11
27 FFFFFFFF, --, --, --, --, MaleArgonian, WERoad09, WERoad09Offer, Topic, PlayerDialogue, Custom, WERoad09OfferTopic, Fo >
```

Figure 1316 - dialogueExport.txt modified.

Close Notepad++. Rename dialogueExport.txt to dialogueExport.csv and open it in Excel.

The screenshot shows a Microsoft Excel spreadsheet titled "dialogueExport.csv - Excel". The data is organized into 17 rows and 14 columns. The columns are labeled: LINE ID, SPEAKER, NPCID, GetISID, RACE, VOICE, TYF, QUEST, BRANCH, CATEGORY, TYPE, SUBTYPE, TOPIC, TOPICINFO, and RESPONSE. The first row is highlighted in green, indicating it is the header. The data includes various creature and character voices, such as Werewolves, Foxes, and Chickens, across different branches and categories like Combat, Death, Hit, and SharedInfo.

Figure 1317 - dialogueExport.csv.

Delete all columns except 'FULL PATH' and 'RESPONSE TEXT'.

The screenshot shows the same Microsoft Excel spreadsheet after deleting unnecessary columns. Now, there are only two columns: "FULL PATH" and "RESPONSE TEXT". The data consists of 17 rows, each containing a full path to a voice file and its corresponding response text. The responses include greetings, congratulations, and acknowledgments.

Figure 1318 - Removed unnecessary columns.

Remove the column headers. Right click on '1' on the left hand side to select row 1, then select Delete.

Next we need to clear empty lines. Click on ‘B’ to select the entirety of column B, then click on the Sort and Filter button and select ‘Sort A to Z’.

When the Sort Warning pops up, select ‘Expand the selection’ then click Sort.

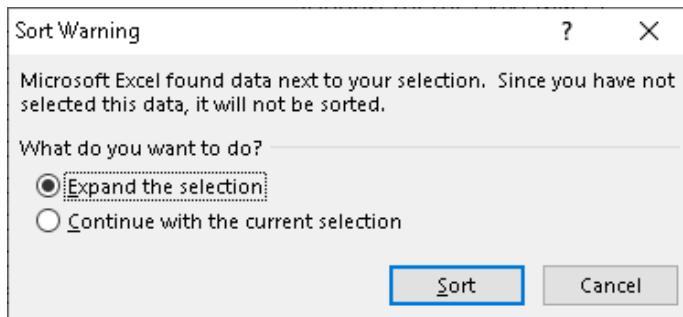


Figure 1319 - Expand the Selection.

Delete the rows where cell B is either blank or has non-dialogue, such as (Deep breath), *cough* or <Laughter>.

Now we need to sort the rows by file name. Click on ‘A’ to select the entirety of column A, then click on the Sort and Filter button and select ‘Sort A to Z’.

Again, when the Sort Warning pops up, select ‘Expand the selection’ then click Sort.

	A	B
1	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_0001CDA9_1.xwm	Don't worry. I'll make sure he sees the light. I trusted you@ now you tr
2	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_0001CDAE_1.xwm	I knew I could trust you!
3	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_000209BB_1.xwm	Vile! None of that - you made a deal@ and the mortal stood by it hone
4	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_000E773E_1.xwm	We should go get the axe.
5	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_000E773F_1.xwm	Let's get the axe and be done with this.
6	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_000E7740_1.xwm	Now that guy was nuts.
7	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_000E7741_1.xwm	Don't bother@ Clavicus won't talk to me.
8	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_000E7742_1.xwm	Wait a second@ there's another option here.
9	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_000E7743_1.xwm	I have a feeling I'm not going to like what happens when we get back to
10	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_000E7744_1.xwm	All this just to kill me Clavicus?
11	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_000E7746_1.xwm	I remember that axe.
12	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_DA03BarabasAboutAx	One of Clavicus's little jests. A wizard named Sebastian Lort had a daug
13	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_DA03BarabasAboutAx	When the daughter became a werewolf it drove Sebastian over the ed
14	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_DA03BarabasAboutAx	The wizard wished for the ability to end his daughter's curse.
15	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_DA03BarabasAboutAx	Clavicus gave him an axe.
16	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_DA03BarbasConvincef	The axe isn't the only item dear old Clavicus has.
17	Data\Sound\Voice\Skyrim.esm\CrDogVoice\DA03_DA03BarbasConvincef	Give him the Rueful Axe and once we're reunited the Masque of Clavic

Figure 1320 - dialogueExport.csv properly sorted.

Save the document then close Excel. Open dialogueExport.csv again in Notepad++.

In Notepad++, go to Search > Replace. In the ‘Find what’ field enter an at symbol ‘@’, and in the ‘Replace with’ field enter a comma ‘,’ then click Replace All.

Next, we need to change the file extensions. In the 'Find what' field enter '.xwm,' and in the 'Replace with' field enter '.wav |' then click Replace All.

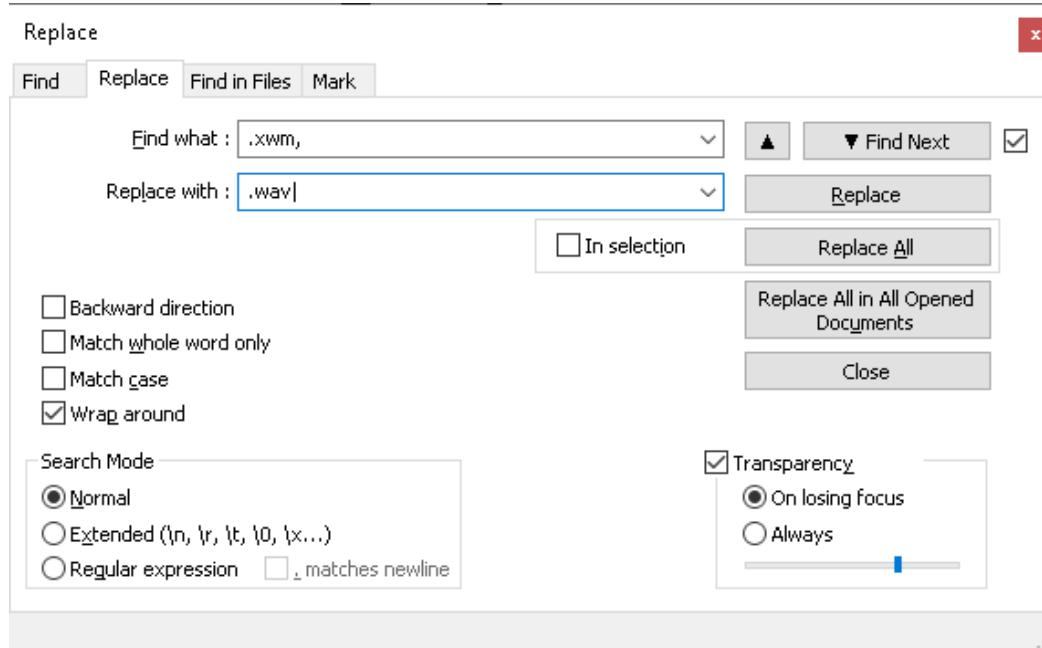


Figure 1321 - Replacing the file extensions and separator character.

We need to make everything lower case. To do this, go to Edit > Select All then go to Edit > Convert Case To > lowercase.

The file paths need to be modified for Google Drive. In the ‘Find what’ field enter ‘data\sound\voice\skyrim.esm\’ and in the ‘Replace with’ field enter ‘/content/drive/MyDrive/’ then click Replace All.

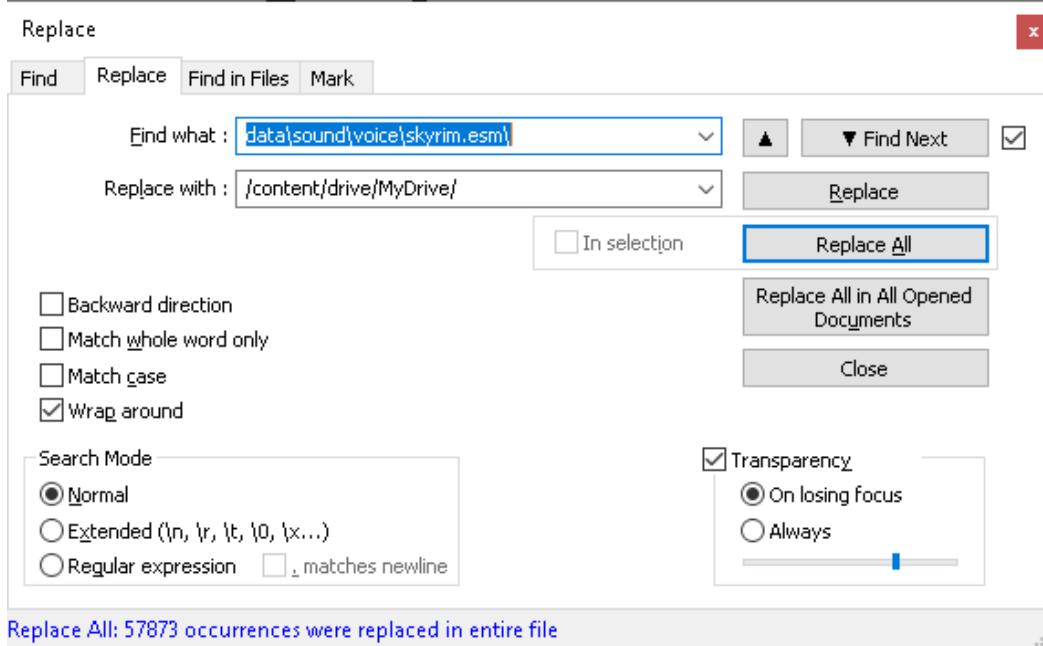


Figure 1322 - Changing the file paths.

Lastly, in the ‘Find what’ field enter ‘\’ and in the ‘Replace with’ field enter ‘/wavs/’ then click Replace All.

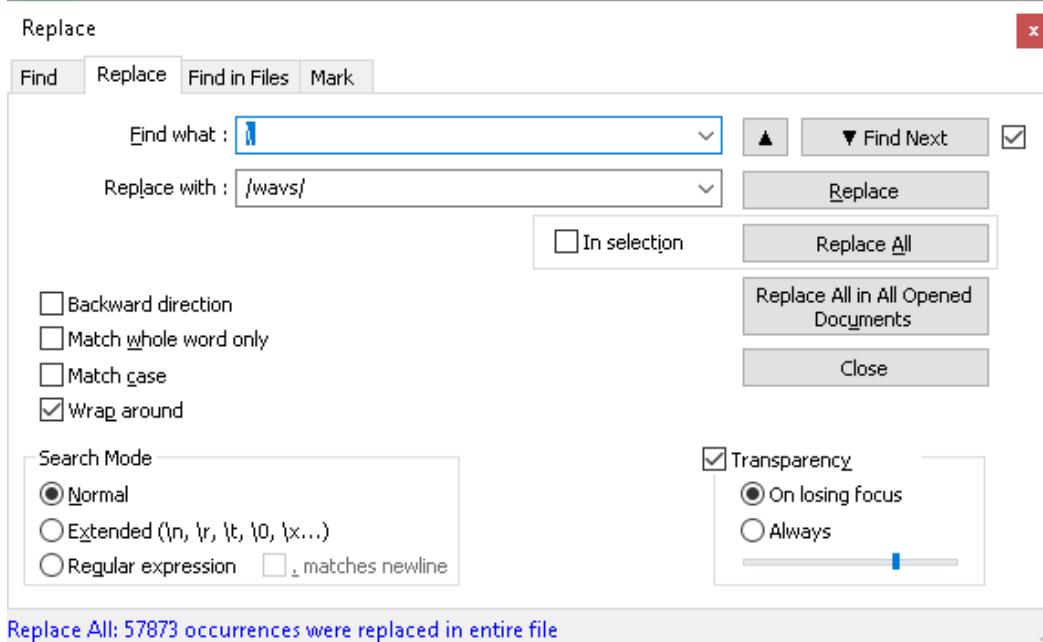


Figure 1323 - Adding wavs folder to file path.

We now have a list of subtitles grouped by order of voice type from which we can easily make new training and validation files.

```

13 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barabasabouttaxeto_000e7693_2.wav|when the daughter became ^
14 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barabasabouttaxeto_000e7693_3.wav|the wizard wished for the
15 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barabasabouttaxeto_000e7693_4.wav|clavicus gave him an axe.
16 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasconvincepla_000e7ea3_1.wav|the axe isn't the only it
17 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasconvincepla_000e7ea3_2.wav|give him the rueful axe a
18 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasfollowtoggel_0001c4df_1.wav|lead the way.
19 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasfollowtoggel_0001c4ed_1.wav|all right, then. we'll ge
20 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasforcegreet0_0007eaed_1.wav|you are exactly what i wa
21 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasgreet_0001c4d9_1.wav|yes, i think you'll be just wha
22 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasgreetcont0_000e0d48_1.wav|you see, my name is barba
23 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasgreeting0_0001bfc0_2.wav|skyrim is now host to giant
24 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasgreeting1a_0001bfc2_3.wav|i know, i know... wars to
25 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasmoreinfo0_0001cdaa_1.wav|well... i guess you could s
26 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasmoreinfo0_0001cdaa_2.wav|he couldn't just kill me, y
27 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasmoreinfo0_0001cdaa_3.wav|of course, because of our s
28 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasoffer0_0001bfc3_1.wav|my master and i had a bit of a
29 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasoffer0_0001bfc3_2.wav|he's kicked me out until i fin
30 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbasoffer1a_0001c4dc_1.wav|very funny. my master is clav
31 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbastovile_0001c4db_1.wav|thank you. now, since he banis
32 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbastovile_0001c4db_2.wav|i know there's a cult that wor
33 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbastovile_0001c4db_3.wav|if this works out, i'll make s
34 /content/drive/MyDrive/crdogvoice/wavs/da03_da03barbaswalkaway_000896a7_1.wav|listen, when you're ready to
35 /content/drive/MyDrive/crdogvoice/wavs/da03_da03hello_0001c4d5_1.wav|well, well. hello there, big man.
36 /content/drive/MyDrive/crdogvoice/wavs/da03_da03hello_0001c4d6_1.wav|hello there, miss.
37 /content/drive/MyDrive/crdogvoice/wavs/da03_da03hello_0001c4d7_1.wav|barbas, at your service.
38 /content/drive/MyDrive/crdogvoice/wavs/da03_da03hello_0001cdb5_1.wav|woof.
39 /content/drive/MyDrive/crdogvoice/wavs/da03_da03hello_0001cdb6_1.wav|woof.

```

Figure 1324 - dialogueExport.csv formatted for Google Colab.

I've uploaded a couple pre-sanitized dialogueExport files from different Bethesda Game Studio titles to make it easier for you to set up new training and validation files.

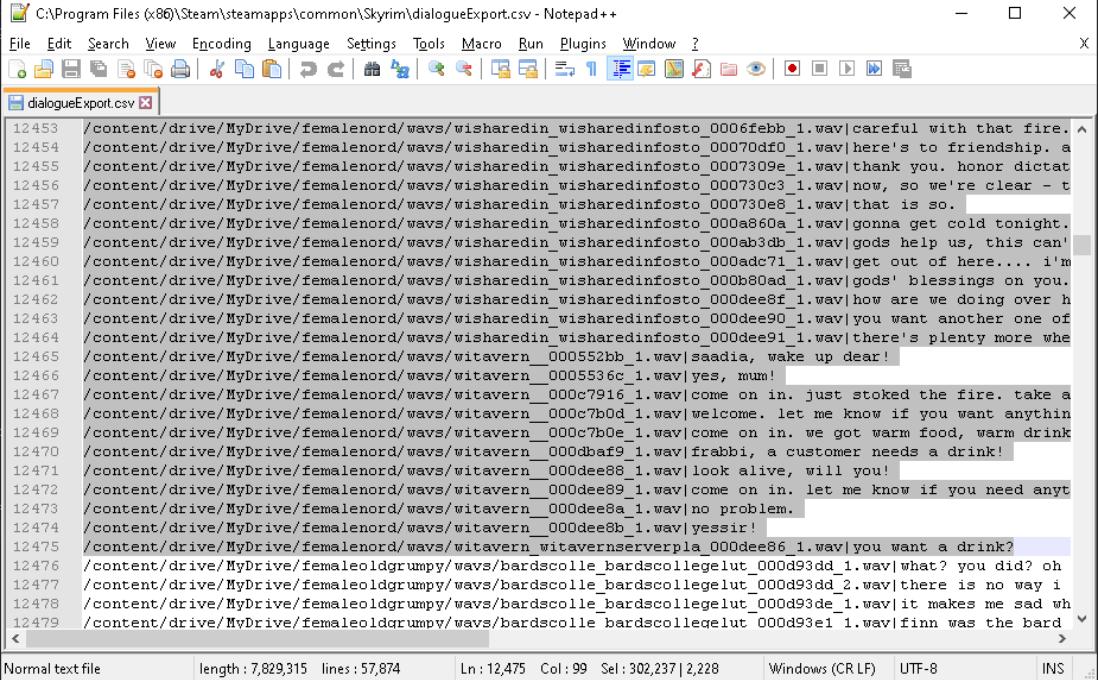
Skyrim:

<https://drive.google.com/file/d/1lqhCZWOoOsWwK6VDIVakPnUmyMS66As1/view?usp=sharing>

Oblivion:

<https://drive.google.com/file/d/1uxbN0NwZcWfeqSYzRoFBLQt4iXZNv5cJ/view?usp=sharing>

Next we need to create a training text file. Select all the voice lines for the specific voice type you plan on training.



```

C:\Program Files (x86)\Steam\steamapps\common\Skyrim\dialogueExport.csv - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
dialogueExport.csv

12453 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_0006febb_1.wav|careful with that fire.
12454 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_00070df0_1.wav|here's to friendship. a
12455 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_0007309e_1.wav|thank you. honor dictat
12456 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_000730c3_1.wav|now, so we're clear - t
12457 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_000730e8_1.wav|that is so.
12458 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_000a860a_1.wav|gonna get cold tonight.
12459 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_000ab3db_1.wav|gods help us, this can't
12460 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_000adc71_1.wav|get out of here.... i'm
12461 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_000b80ad_1.wav|gods' blessings on you.
12462 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_000dee8f_1.wav|how are we doing over h
12463 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_000dee90_1.wav|you want another one of
12464 /content/drive/MyDrive/femalenord/wavs/wisharedin_wisharedinfosto_000dee91_1.wav|there's plenty more whe
12465 /content/drive/MyDrive/femalenord/wavs/witavern_000552bb_1.wav|saadia, wake up dear!
12466 /content/drive/MyDrive/femalenord/wavs/witavern_0005536c_1.wav|yes, mum!
12467 /content/drive/MyDrive/femalenord/wavs/witavern_000c7916_1.wav|come on in. just stoked the fire. take a
12468 /content/drive/MyDrive/femalenord/wavs/witavern_000c7b0d_1.wav|welcome. let me know if you want anythin
12469 /content/drive/MyDrive/femalenord/wavs/witavern_000c7b0e_1.wav|come on in. we got warm food, warm drink
12470 /content/drive/MyDrive/femalenord/wavs/witavern_000dbaf9_1.wav|frabbit, a customer needs a drink!
12471 /content/drive/MyDrive/femalenord/wavs/witavern_000dee88_1.wav|look alive, will you!
12472 /content/drive/MyDrive/femalenord/wavs/witavern_000dee89_1.wav|come on in. let me know if you need anyt
12473 /content/drive/MyDrive/femalenord/wavs/witavern_000dee8a_1.wav|no problem.
12474 /content/drive/MyDrive/femalenord/wavs/witavern_000dee8b_1.wav|yessir!
12475 /content/drive/MyDrive/femalenord/wavs/witavern_witavernserverpla_000dee86_1.wav|you want a drink?
12476 /content/drive/MyDrive/femaleoldgrumpy/wavs/bardscolle_bardscollegegelut_000d93dd_1.wav|what? you did? oh
12477 /content/drive/MyDrive/femaleoldgrumpy/wavs/bardscolle_bardscollegegelut_000d93dd_2.wav|there is no way i
12478 /content/drive/MyDrive/femaleoldgrumpy/wavs/bardscolle_bardscollegegelut_000d93de_1.wav|it makes me sad wh
12479 /content/drive/MyDrive/femaleoldgrumpy/wavs/bardscolle_bardscollegegelut_000d93e1_1.wav|finn was the bard

```

Figure 1325 - Select all lines of dialogue for a specific voice type.

Create a new text file and name it '#_training.txt' where '#' is the name of the voice type you're training, i.e.: femalenord. In my example that's going to be 'femalenord_training.txt' because I'll be making a dataset based on the femalenord voice type.

Copy the lines you highlighted from dialogueExport.csv and paste them into your #_training.txt file.

Now highlight at least 50 lines from your training text file.

```

31 /content/drive/MyDrive/femalenord/wavs/cw02a_mq103asoldierblockin_00059eec_1.wav|nobody allowed in. ulfric ^
32 /content/drive/MyDrive/femalenord/wavs/cw02b_0006f970_1.wav|damn dustmen! why don't you stay dead?
33 /content/drive/MyDrive/femalenord/wavs/cw02b_0006fb15_1.wav|look out! draugr!
34 /content/drive/MyDrive/femalenord/wavs/cw02b_000a0948_1.wav|no. and i'm not sure i'm better off for it no
35 /content/drive/MyDrive/femalenord/wavs/cw02b_000a0de0_1.wav|what in the nine holds is that?
36 /content/drive/MyDrive/femalenord/wavs/cw02b_000a0de3_1.wav|but there isn't any other way through.
37 /content/drive/MyDrive/femalenord/wavs/cw02b_000a0df3_1.wav|oh... i've heard of this. they say these wall
38 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059eed_1.wav|you must be the new unblo
39 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059eee_1.wav|you must be the new unblo
40 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059ef1_1.wav|i don't know what we're d
41 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059ef0_1.wav|i keep thinking i see som
42 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059ef1_1.wav|i hope we don't run into
43 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059ef2_1.wav|what are we waiting for?
44 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_0005a086_1.wav|keep a sharp eye out.
45 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_000a094e_1.wav|at least we got the damn
46 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_000a094f_1.wav|i wish galmar would hurry
47 /content/drive/MyDrive/femalenord/wavs/cwattackci_cwattackcityblo_0002088e_1.wav|they will sing stories of
48 /content/drive/MyDrive/femalenord/wavs/cwattackcity_0001a2e0_1.wav|aye!
49 /content/drive/MyDrive/femalenord/wavs/cwdialogue_cwdialoguehello_00020956_1.wav|i long to be out there, w
50 /content/drive/MyDrive/femalenord/wavs/cwdialogue_cwdialoguehello_00020958_1.wav|legion soldiers gleam lik
51 /content/drive/MyDrive/femalenord/wavs/cwdialogue_cwdialoguehello_0002095b_1.wav|i'll fight the entire leg
52 /content/drive/MyDrive/femalenord/wavs/cwdialoguesoldiers01_0006b139_1.wav|hope i don't draw guard duty.
53 /content/drive/MyDrive/femalenord/wavs/cwdialoguesoldiers01_0006b13a_1.wav|i bet we have rabbit again. i'
54 /content/drive/MyDrive/femalenord/wavs/cwdialoguesoldiers01_0006b13b_1.wav|we're running out of bread. i
55 /content/drive/MyDrive/femalenord/wavs/cwdialoguesoldiers01_0006b145_1.wav|you'd rather be plowing a fiel
56 /content/drive/MyDrive/femalenord/wavs/cwdialoguesoldiers01_0006b146_1.wav|go tell the commander. i'm sur
57 /content/drive/MyDrive/femalenord/wavs/cwdialoguesoldiers01_0006b147_1.wav|you must be all right if you've

```

Figure 1326 - 50 lines highlighted.

Create another text file and name it '#_validation.txt' where '#' is the name of the voice type you're training, i.e.: femalenord. In my example that's going to be 'femalenord_validation.txt'.

Cut the highlighted lines out of the training text file and paste them into the validation text file.

```

25 /content/drive/MyDrive/femalenord/wavs/cw02a_0003a096_1.wav|maybe they're so scared of you they ran away. ^
26 /content/drive/MyDrive/femalenord/wavs/cw02a_0006f964_1.wav|it's the legion!
27 /content/drive/MyDrive/femalenord/wavs/cw02a_0006f965_1.wav|imperials! kill them!
28 /content/drive/MyDrive/femalenord/wavs/cw02a_0006fb0f_1.wav|it's got to be the imperials! find them!
29 /content/drive/MyDrive/femalenord/wavs/cw02a_0007641b_1.wav|the legion is here! defend the entrance!
30 /content/drive/MyDrive/femalenord/wavs/cw02a_000d6619_1.wav|just shut up and keep out of sight.
31 /content/drive/MyDrive/femalenord/wavs/cw02a_mq103asoldierblockin_00059eec_1.wav|nobody allowed in. ulfric
32 /content/drive/MyDrive/femalenord/wavs/cw02b_0006f970_1.wav|damn dustmen! why don't you stay dead?
33 /content/drive/MyDrive/femalenord/wavs/cw02b_0006fb15_1.wav|look out! draugr!
34 /content/drive/MyDrive/femalenord/wavs/cw02b_000a0948_1.wav|no. and i'm not sure i'm better off for it no
35 /content/drive/MyDrive/femalenord/wavs/cw02b_000a0de0_1.wav|what in the nine holds is that?
36 /content/drive/MyDrive/femalenord/wavs/cw02b_000a0de3_1.wav|but there isn't any other way through.
37 /content/drive/MyDrive/femalenord/wavs/cw02b_000a0df3_1.wav|oh... i've heard of this. they say these wall
38 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059eed_1.wav|you must be the new unblo
39 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059eee_1.wav|you must be the new unblo
40 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059ef1_1.wav|i don't know what we're d
41 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059ef0_1.wav|i keep thinking i see som
42 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059ef1_1.wav|i hope we don't run into
43 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_00059ef2_1.wav|what are we waiting for?
44 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_0005a086_1.wav|keep a sharp eye out.
45 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_000a094e_1.wav|at least we got the damn
46 /content/drive/MyDrive/femalenord/wavs/cw02b_mq103bsoldierblockin_000a094f_1.wav|i wish galmar would hurry
47 /content/drive/MyDrive/femalenord/wavs/cwattackci_cwattackcityblo_0002088e_1.wav|they will sing stories of
48 /content/drive/MyDrive/femalenord/wavs/cwattackcity_0001a2e0_1.wav|aye!
49 /content/drive/MyDrive/femalenord/wavs/cwdialogue_cwdialoguehello_00020956_1.wav|i long to be out there, w
50 /content/drive/MyDrive/femalenord/wavs/cwdialogue_cwdialoguehello_00020958_1.wav|legion soldiers gleam lik

```

Figure 1327 - Validation text file.

Save both your training and validation text files.

Now that we've prepared the training and validation files, we can move on to preparing the audio.

Extract the voice acting from the 'Skyrim - Voices.bsa' archive. To do this we'll need a copy of [BAE](#), the Bethesda Archive Extractor, which you can download from Nexusmods.

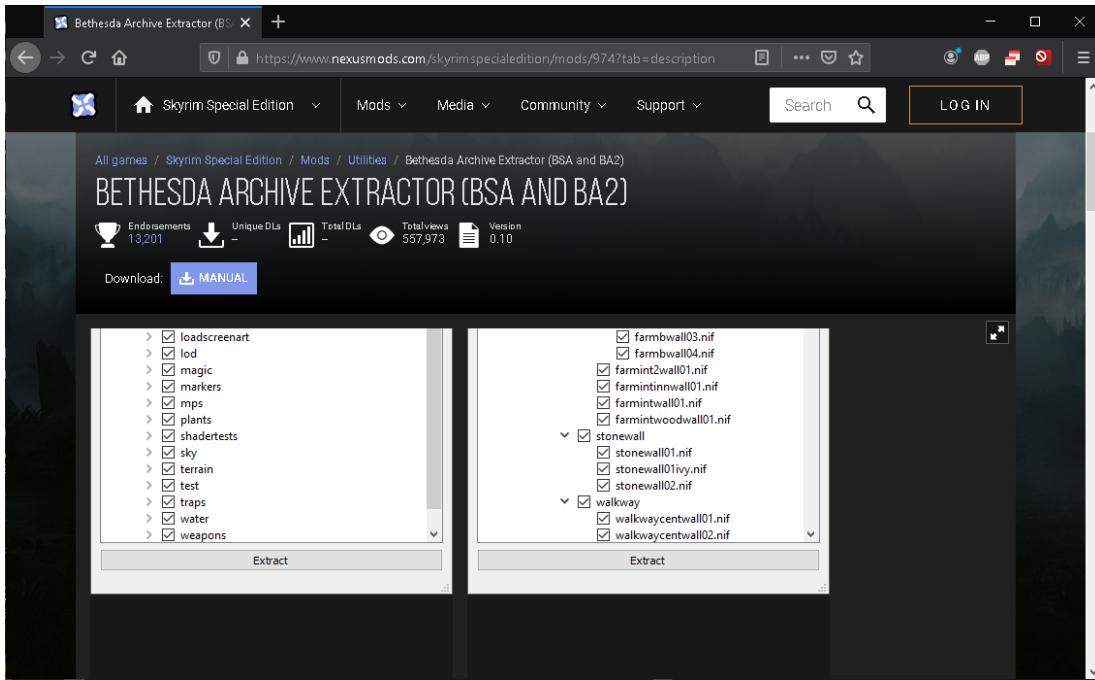


Figure 1328 - BAE on Nexusmods.

Extract the archive to a folder and run bae.exe.

Drag and drop the ‘Skyrim - Voices.bsa’ file into BAE to open it.

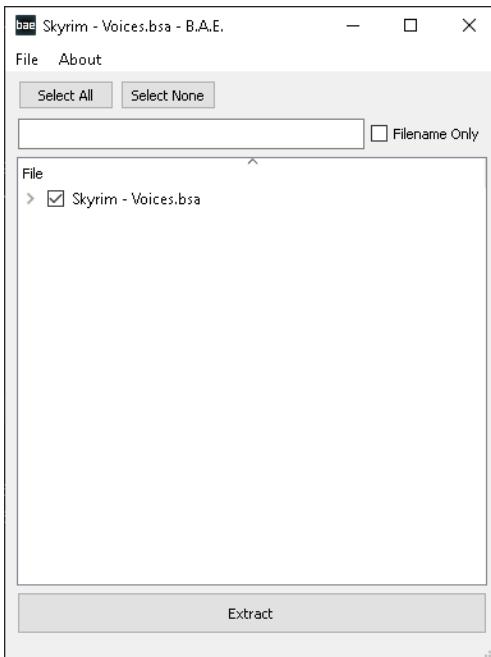


Figure 1329 - Skyrim - Voices.bsa opened in BAE.

Click ‘Select All’ to ensure everything is selected, then click Extract.

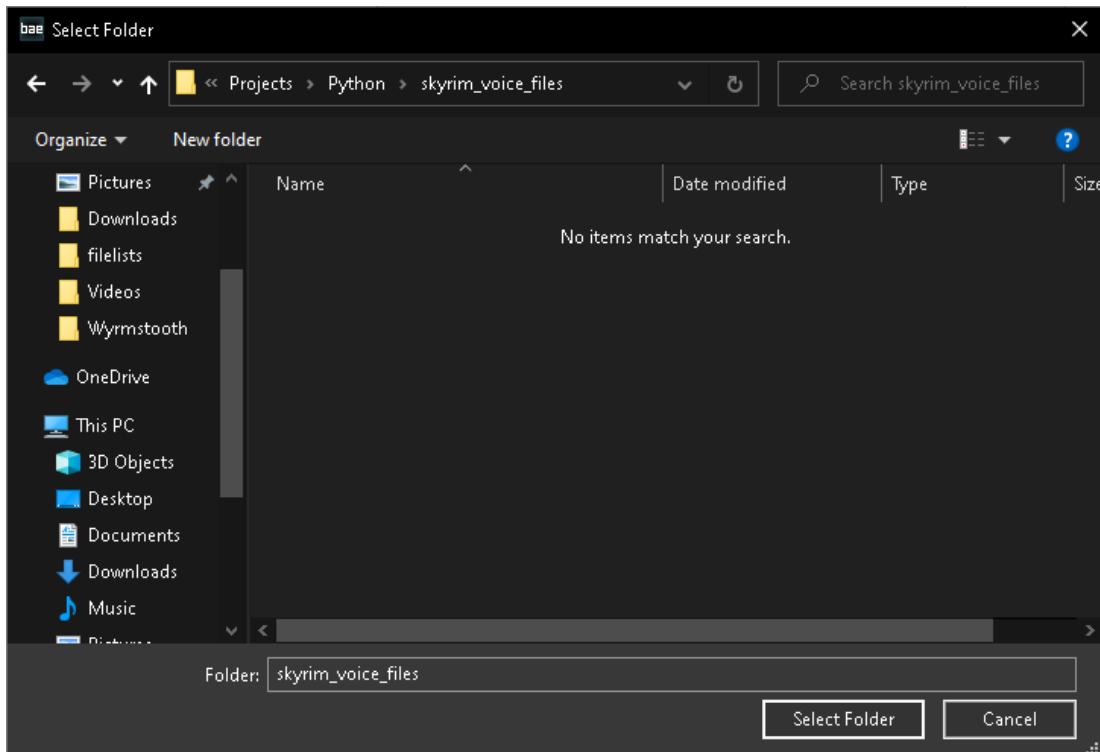


Figure 1330 - Selecting a folder to extract to.

Select a folder to extract the voice files to, then click on Select Folder.

Once BAE has finished extracting everything from the ‘Skyrim - Voices.bsa’ archive, navigate to the folder you selected, then go to sound > voice > skyrim.esm. You should see a folder for each voice type and these folders should contain .fuz files.

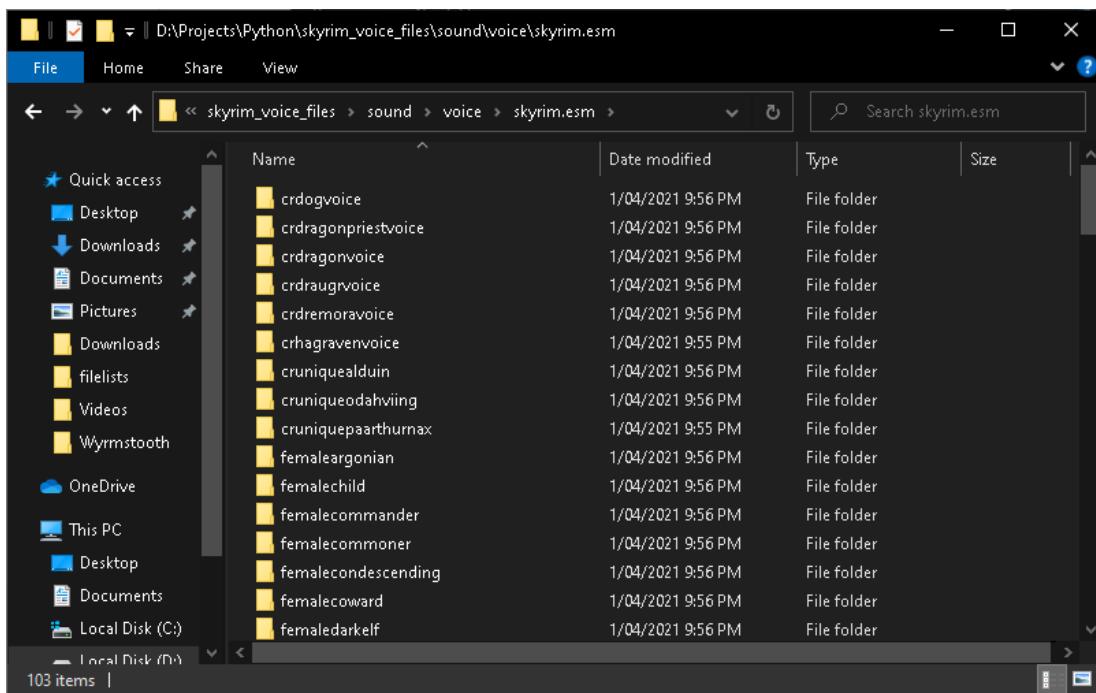


Figure 1331 - Extracted voice type folders.

We'll need to convert those .fuz files to .wav next.

Go ahead and download [Unfuzer](#) from Nexusmods.

Extract the archive to a folder and run unfuzer.exe.

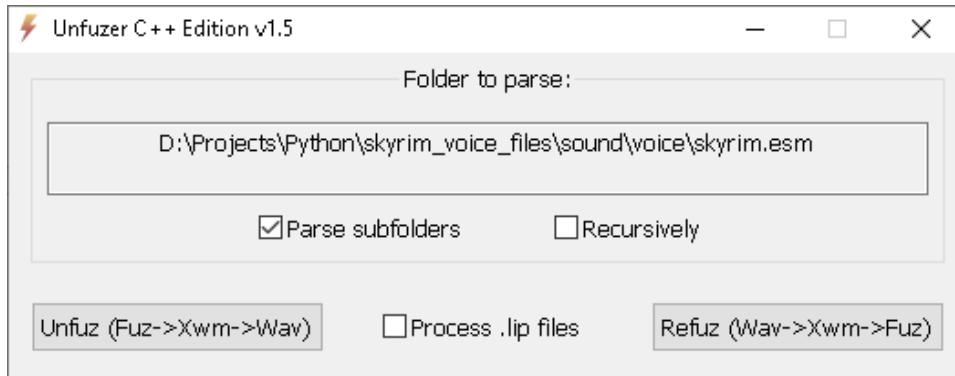


Figure 1332 - Unfuzer.

Click on the file path beneath 'Folder to parse'. Navigate to the folder where you extracted the voice files to, go to sound > voice and select the 'skyrim.esm' folder.

Tick 'Parse subfolders', make sure 'Process .lip Files' isn't ticked, then click Unfuz (Fuz->Xwm->Wav). This will convert every .fuz file in the skyrim.esm subfolders to .wav.

Note: Converting all the .fuz files to .wav will take some time.

In the ‘skyrim.esm’ folder create a new text file. Rename that file ‘delete_fuz.bat’

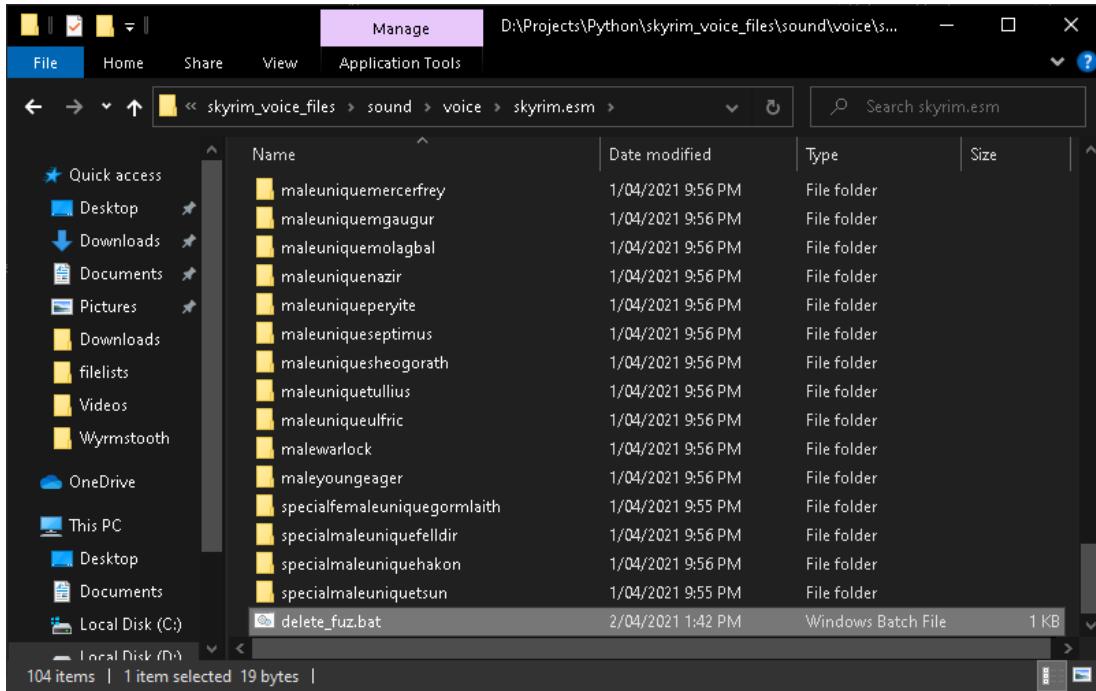


Figure 1333 - Created delete_fuz.bat.

Open the batch file in Notepad and enter the following commands:

```
del /S *.fuz
pause
```



Figure 1334 - delete_fuz.bat contents.

Save and close Notepad then run the batch file. This will delete all .fuz files from the subfolders leaving only the .wav files behind.

We'll need to create a project directory for this next part. I just made a folder called Datasets but you can name it whatever you like.

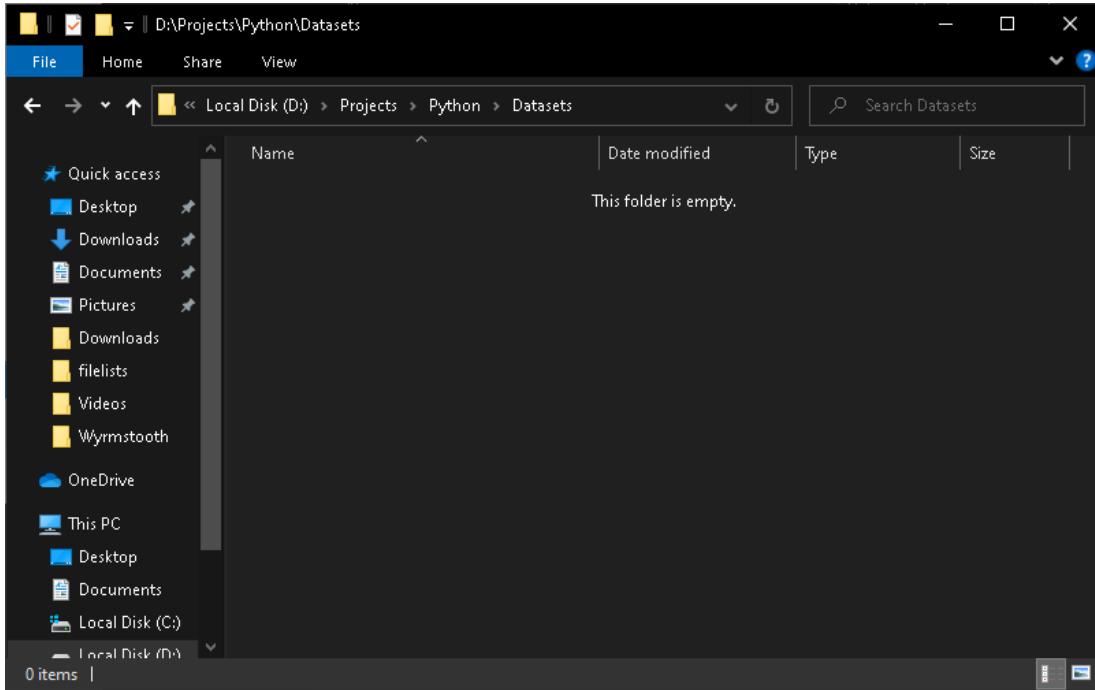


Figure 1335 - New project directory for setting up our dataset.

Copy the folder containing the .wav files for your chosen voice type over to your project folder. In my example I'll be copying the 'femalenord' folder from the 'skyrim.esm' folder over to the Datasets folder I just created.

Once you've copied it to the project folder, append '_base' to the folder name.

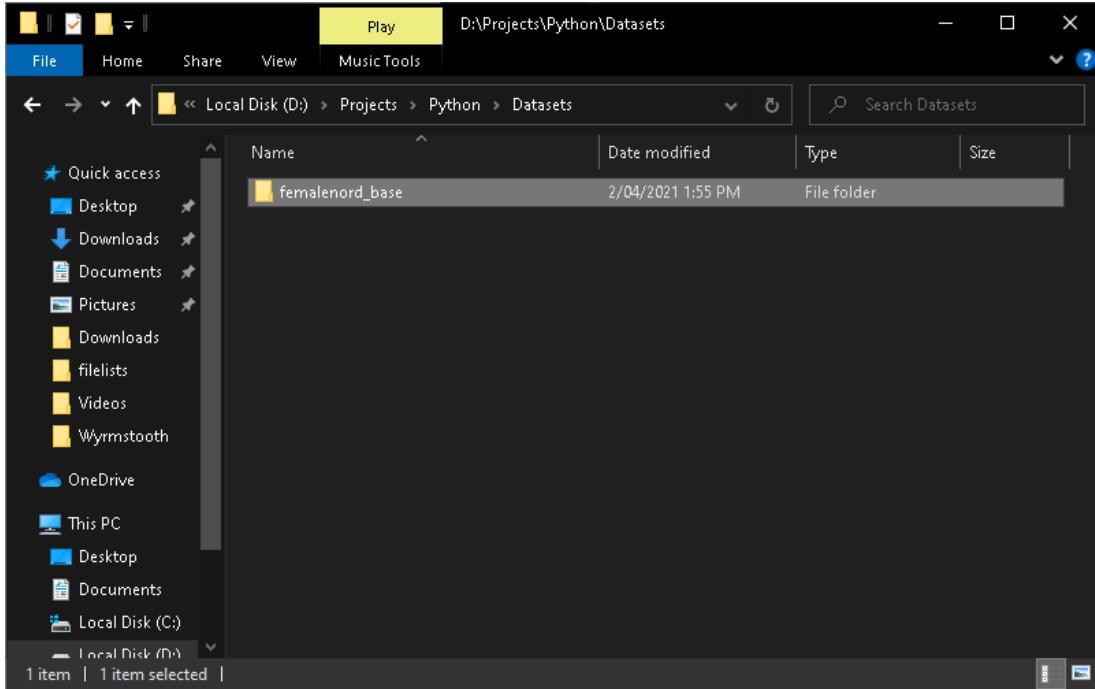


Figure 1336 - Copied the unprocessed audio files to the project folder.

Tacotron and WaveGlow will require audio files to be formatted in a specific way, so in this section I'll be showing you how to batch process the audio files for your dataset. In order to do so we'll need to download two utilities: SoX and ffmpeg.

[SoX](#) can be downloaded from its page on Sourceforge: <http://sox.sourceforge.net/>.

A screenshot of a web browser displaying the SoX - Sound eXchange | HomePage. The URL in the address bar is 'sox.sourceforge.net'. The page features a logo of a speaker with sound waves. On the left, there's a sidebar with links: Home Page, Features, FAQ, Documentation, Mailing Lists, Links, Project Page, and Browse git. Below that are buttons for 'Support this project' and 'Hosted by SOURCEFORGE'. The main content area has a 'Welcome' header and a paragraph about SoX being a cross-platform command line utility for sound processing. To the right, there's a code block showing a command-line session demonstrating SoX processing a WAV file into a FLAC file, including various effects like remixing, normalizing, and applying highpass filters. At the bottom of the code block is a link: 'For the list of all file formats, device drivers, &c.'

Figure 1337 - SoX website.

Once downloaded, extract the archive to an easily accessible location. On my end I just extracted SoX to a new folder in Program Files (x86).

[ffmpeg](https://www.ffmpeg.org/download.html#build-windows) can be downloaded from its website. Again, once you've downloaded it, extract the files to an easily accessible location. On my end I just extracted ffmpeg to a new folder in Program Files (x86).

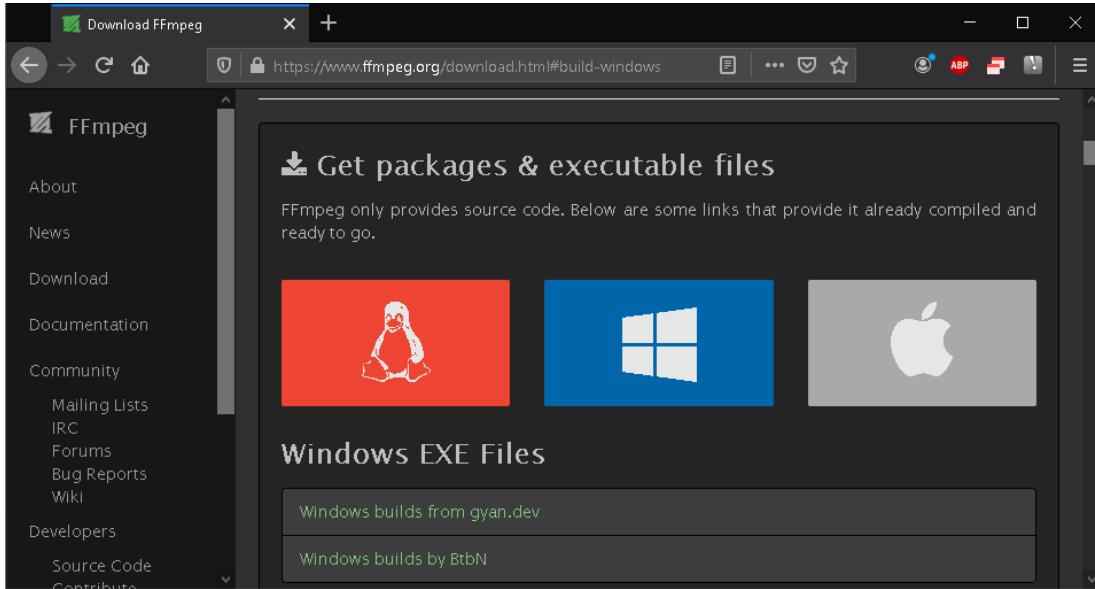


Figure 1338 - ffmpeg website.

Firstly, if you're copying audio files from Oblivion, or from a game where the audio files aren't in .wav format, we'll need to convert them to .wav first.

Create a new batch file and name it '0_mp3_to_wav.bat'. Open it in Notepad and enter in the following commands:

```
cd "D:\Projects\Python\Datasets\femalenord_base\"  
for %a in (*.mp3) do "C:\Program Files (x86)\ffmpeg\bin\ffmpeg.exe"  
-i "%a" "%~na.wav"  
pause
```

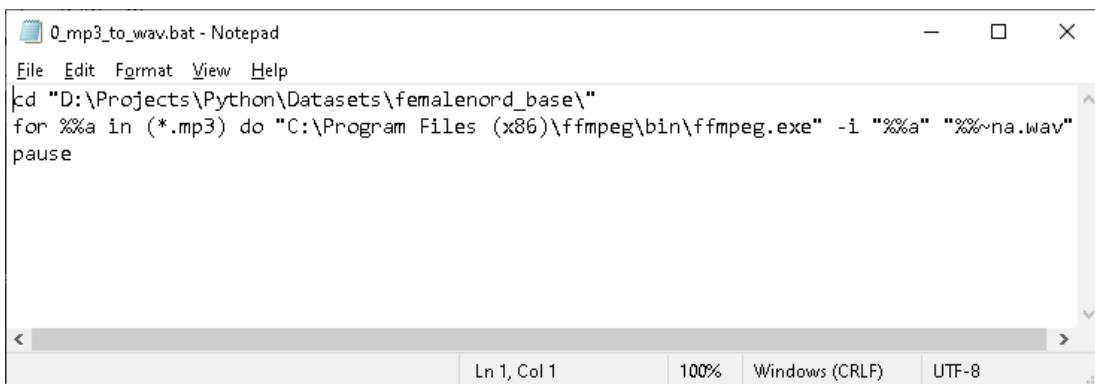


Figure 1339 - 0_mp3_to_wav.bat.

Note: You'll need to change the folder paths to match your environment. In this example I'm converting .mp3 files to .wav. If your files are already in .wav format you won't need to do this.

Save and close Notepad then run ‘0_mp3_to_wav.bat’ and wait for it to complete. The .wav files will be outputted back to the folder containing the original files.

Now we need to trim silence from our .wav files. To do this, create a new batch file and name it ‘1_trim_silence.bat’. Open it in Notepad and enter in the following commands:

```
if not exist "D:\Projects\Python\Datasets\output_1" mkdir  
D:\Projects\Python\Datasets\output_1  
cd "D:\Projects\Python\Datasets\femalenord_base"  
FOR %%F IN (*.wav) DO "C:\Program Files (x86)\sox-14-4-2\sox.exe"  
%%F "D:\Projects\Python\Datasets\output_1\%%~nxF"^(  
silence 1 0.1 1% reverse silence 1 0.1 1% reverse silence 1 0.1 1%  
reverse  
pause
```

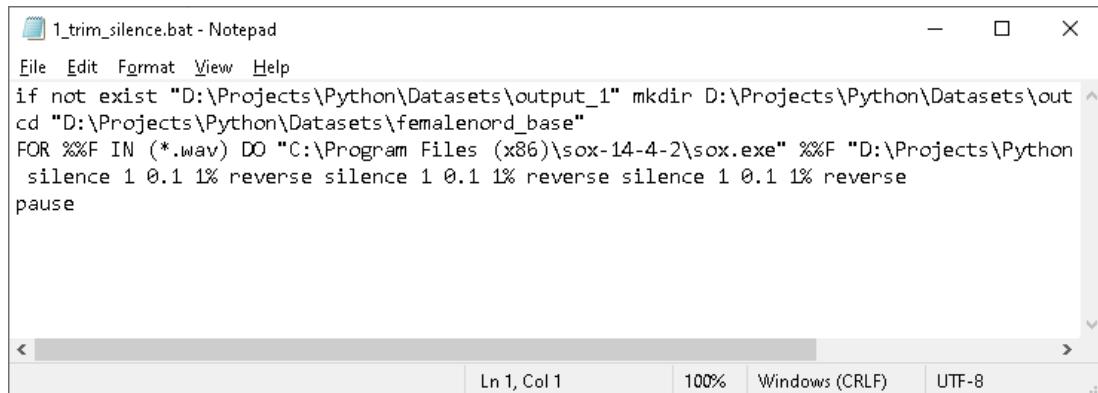


Figure 1340 - 1_trim_silence.bat.

Note: You'll need to change the folder paths to match your environment. We'll be reading .wav files from the #_base folder and outputting to a folder called output_1.

Save and close Notepad then run ‘1_trim_silence.bat’ and wait for it to complete.

Next we'll need to add some silence back to the end of each .wav file. In order to do this, create a new batch file and name it '2_add_silence.bat'. Open it in Notepad and enter in the following commands:

```
if not exist "D:\Projects\Python\Datasets\output_2" mkdir  
D:\Projects\Python\Datasets\output_2  
cd "D:\Projects\Python\Datasets\output_1\"  
FOR %%F IN (*.wav) DO "C:\Program Files (x86)\sox-14-4-2\sox.exe"  
%%F "D:\Projects\Python\Datasets\output_2\%%~nxF" pad 0 0.1  
pause
```

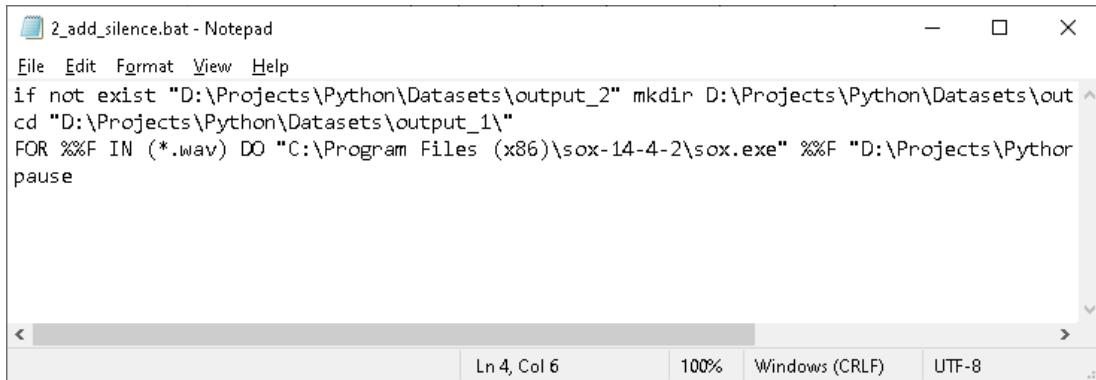


Figure 1341 - 2_add_silence.bat.

Note: You'll need to change the folder paths to match your environment. We'll be reading .wav files from the output_1 folder and outputting to a folder called output_2.

Save and close Notepad then run '2_add_silence.bat' and wait for it to complete.

We'll need to downsample the audio to 22050Hz. Voice acting in most Bethesda Game Studios titles is recorded at 44100Hz. Create a new batch file and name it '3_downsample.bat'. Open it in Notepad and enter in the following commands:

```
if not exist "D:\Projects\Python\Datasets\output_3" mkdir  
D:\Projects\Python\Datasets\output_3  
cd "D:\Projects\Python\Datasets\output_2\"  
FOR %%F IN (*.wav) DO "C:\Program Files (x86)\sox-14-4-2\sox.exe"  
%%F "D:\Projects\Python\Datasets\output_3\%%~nxF" rate -v 22050  
pause
```

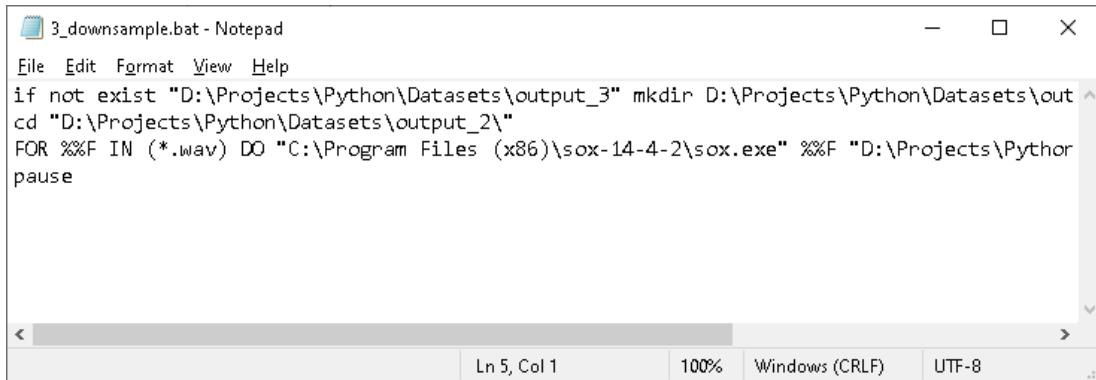


Figure 1342 - 3_downsample.bat.

Note: You'll need to change the folder paths to match your environment. We'll be reading .wav files from the output_2 folder and outputting to a folder called output_3.

Save and close Notepad then run '3_downsample.bat' and wait for it to complete.

Lastly we need to make sure all our .wav files are in mono. Create a new batch file and name it '4_mono.bat'. Open it in Notepad and enter in the following commands:

```
if not exist "D:\Projects\Python\Datasets\output_4" mkdir  
D:\Projects\Python\Datasets\output_4  
cd "D:\Projects\Python\Datasets\output_3\"  
for %%S in (*.wav) do "C:\Program Files (x86)\ffmpeg\bin\ffmpeg.exe"  
-i "%%S" -ac 1 "D:\Projects\Python\Datasets\output_4\%%S"  
pause
```

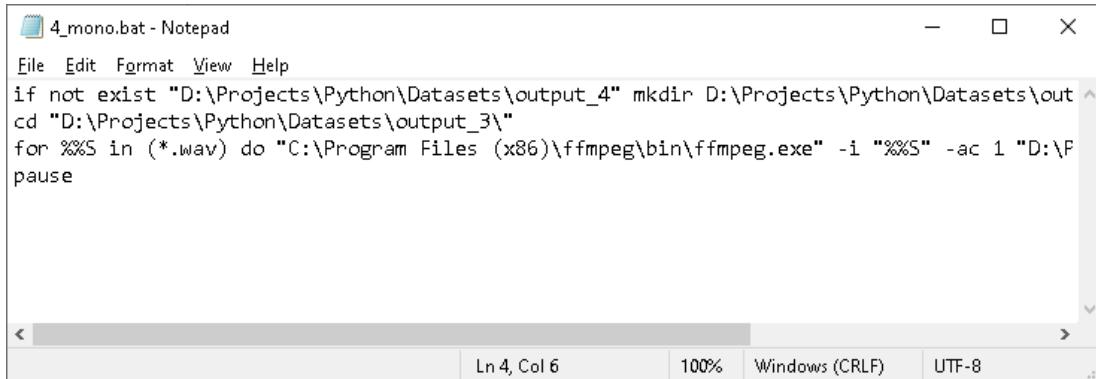


Figure 1343 - 4_mono.bat.

Note: You'll need to change the folder paths to match your environment. We'll be reading .wav files from the output_3 folder and outputting to a folder called output_4.

Save and close Notepad then run '4_mono.bat' and wait for it to complete.

Rename the folder output_4 to match the name of the voice type you're setting up a dataset for. In my example, that will be femalenord.

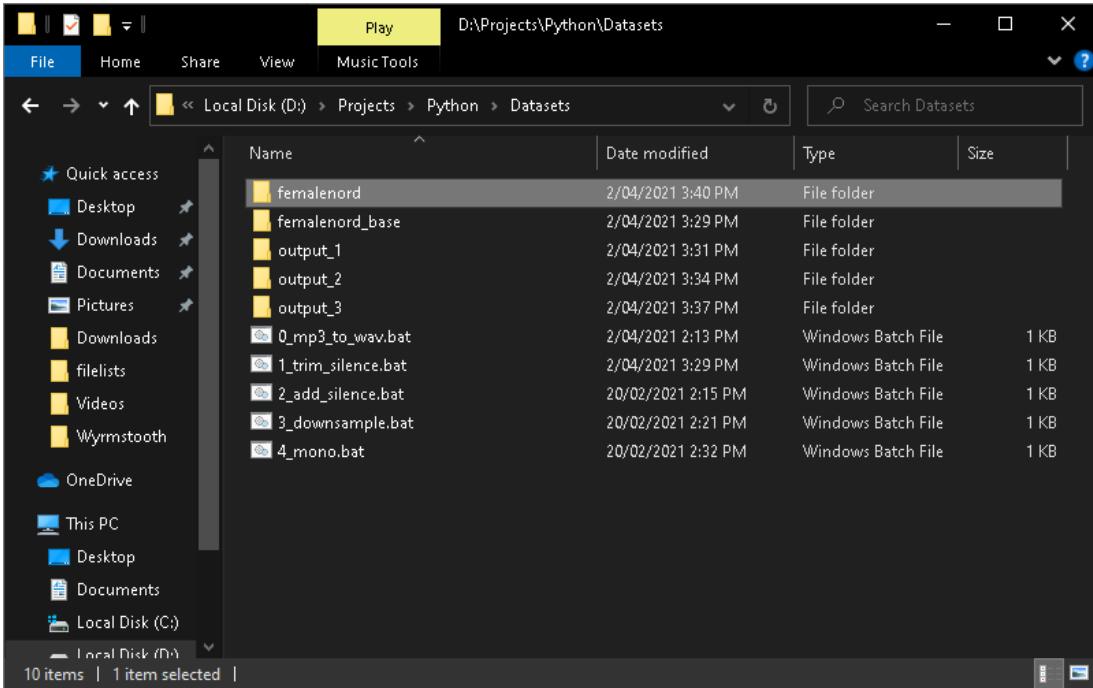


Figure 1344 - Renamed the output_4 folder.

Create a new folder named ‘wavs’ within it, then move all the .wav files into the ‘wavs’ folder.

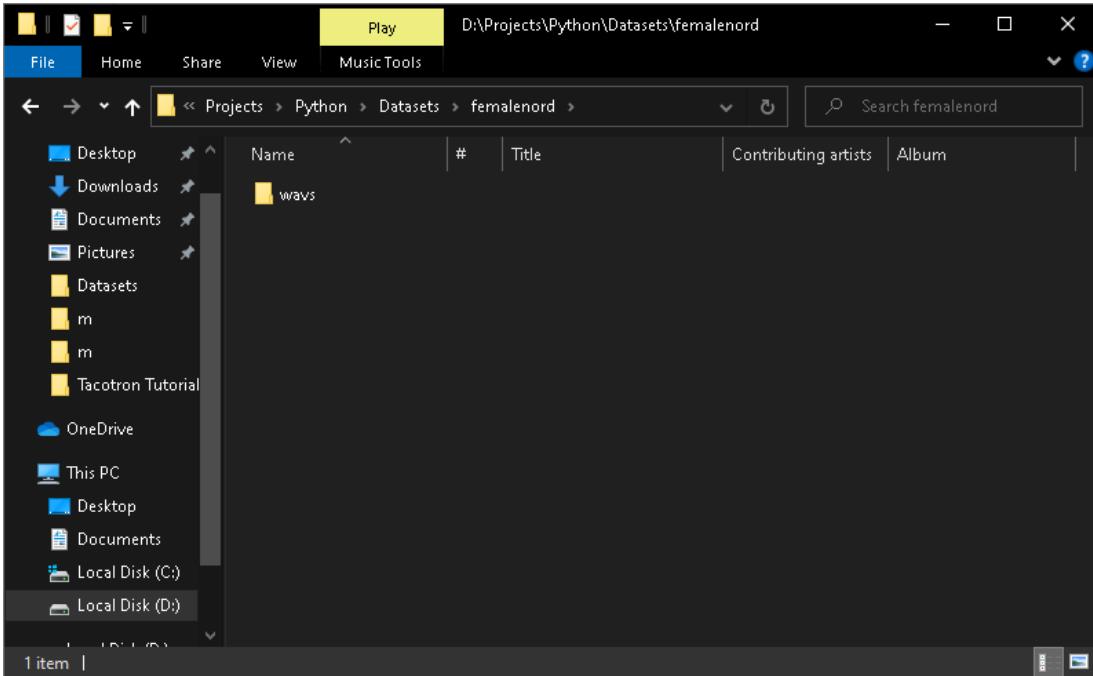


Figure 1345 - wavs folder.

Go to your Google Drive: <https://drive.google.com/>.

Create a new folder in Google Drive called ‘filelists’, then drag-and-drop the training and validation text files into your browser to upload them. In my example that’s going to be the femalenord_training.txt and femalenord_validation.txt files.

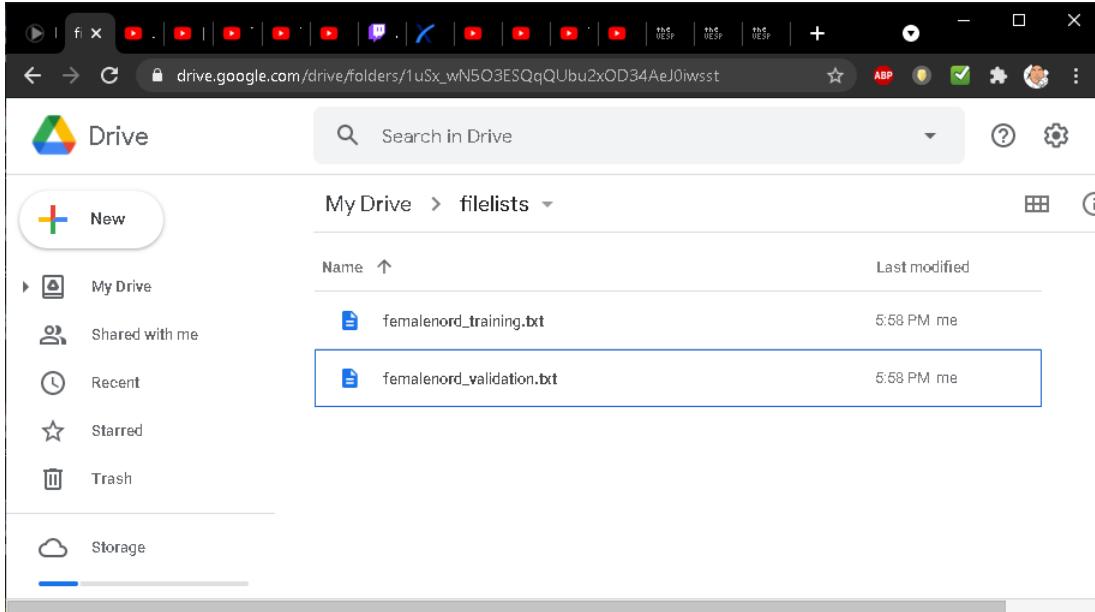


Figure 1346 - Uploading the training and validation files.

Drag-and-drop the folder containing the processed .wav files into your browser to the root of your Google Drive directory to upload them. In my example that’s going to be the femalenord folder.

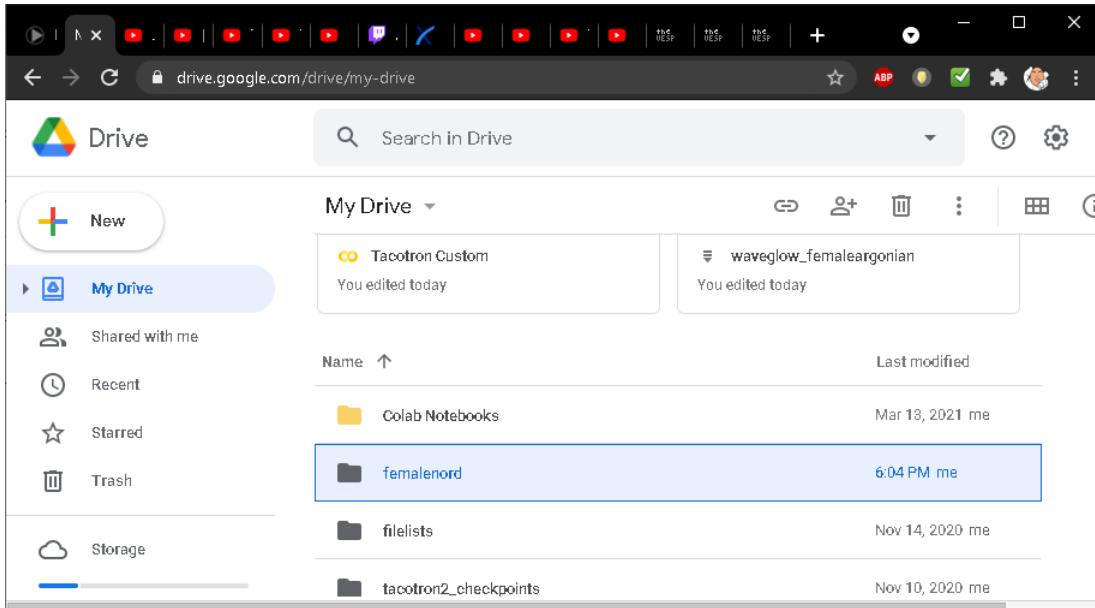


Figure 1347 - Uploading the processed audio files.

MOUNTING YOUR GOOGLE DRIVE STORAGE IN GOOGLE COLAB

As mentioned earlier, I've already set up a Colab Notebook with all the commands you'll need to run. You can access it here:

<https://colab.research.google.com/drive/13vRLNPLqVWGigHGuUjJKxXuZw93PUBBr?usp=sharing>

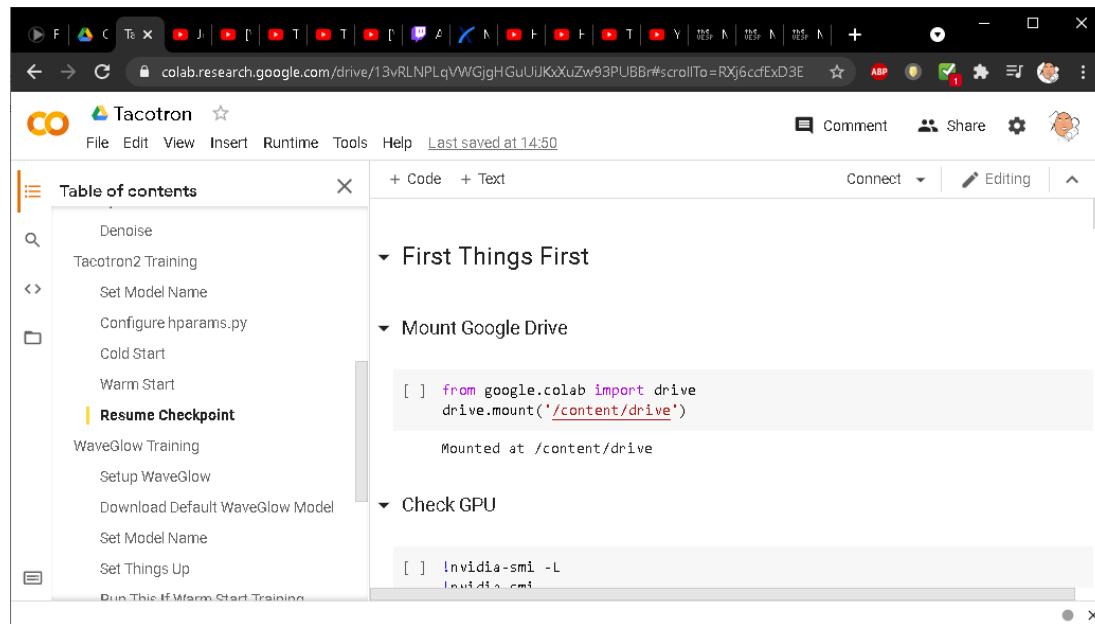


Figure 1348 - Colab Notebook.

The first thing we need to do is go up to Runtime > Change Runtime Type.

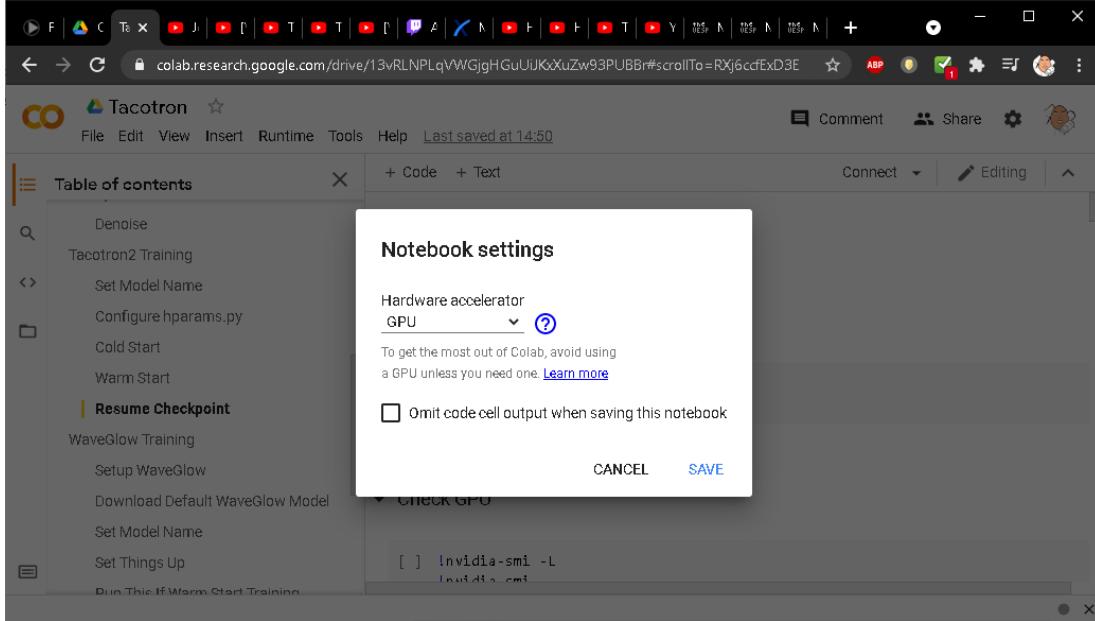


Figure 1349 - Notebook settings.

Make sure 'Hardware accelerator' is set to GPU then click Save.

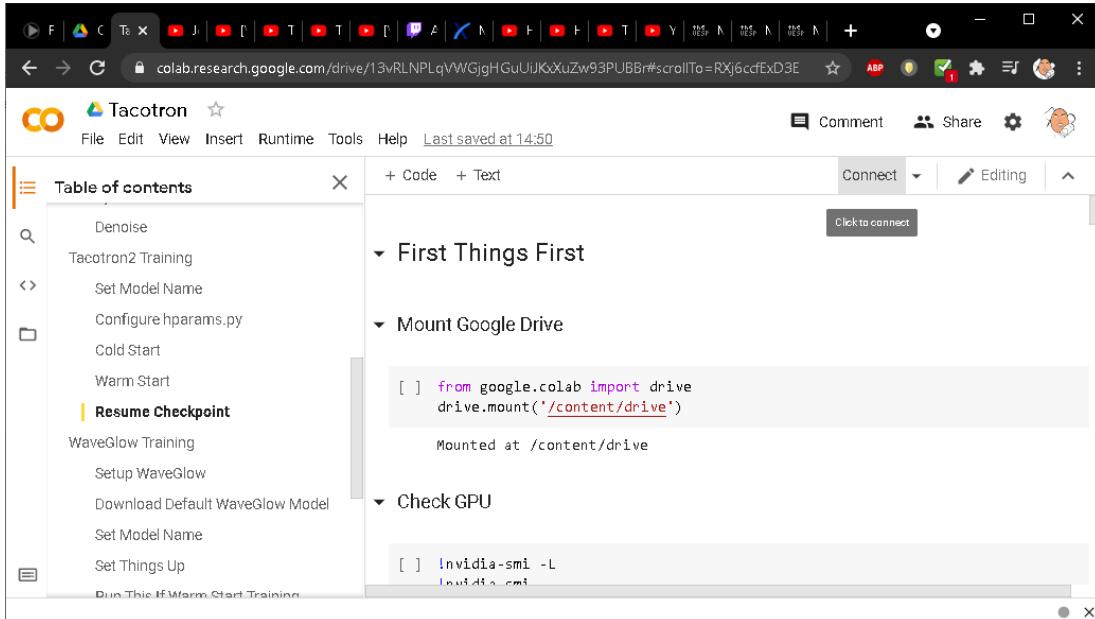


Figure 1350 - Clicking Connect.

Click Connect near the top-right to start a new Colab session.

It's important to note that sessions are temporary. Anything we install or save to our session will be lost once the session is terminated. We can terminate sessions manually by clicking on the down arrow next to the Connect button, selecting Manage Sessions, then click Terminate next to our active session.

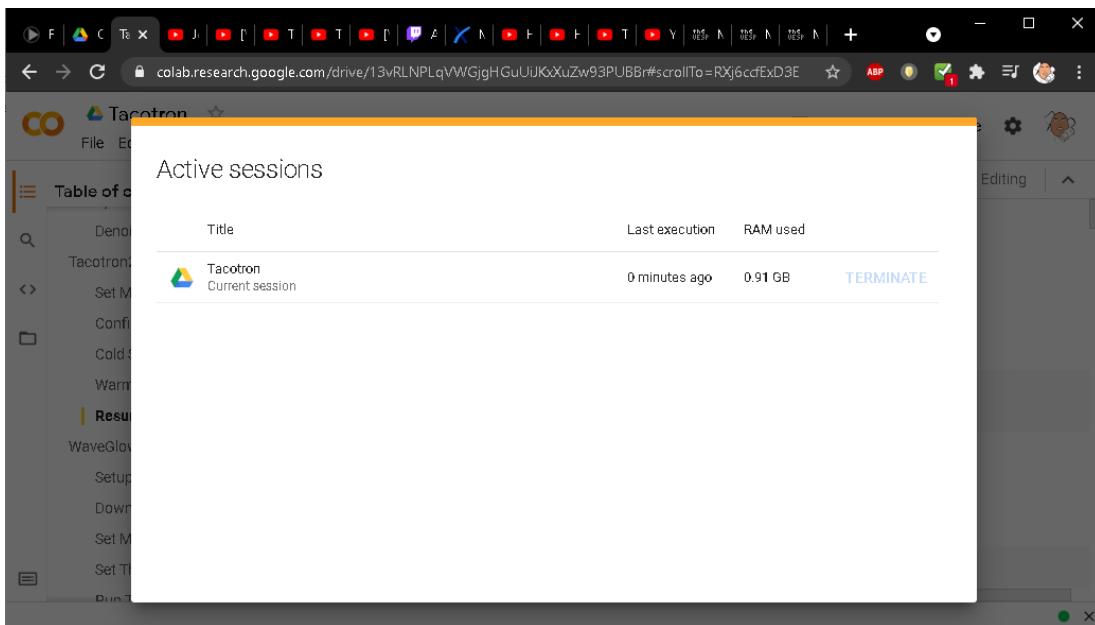


Figure 1351 - Active session.

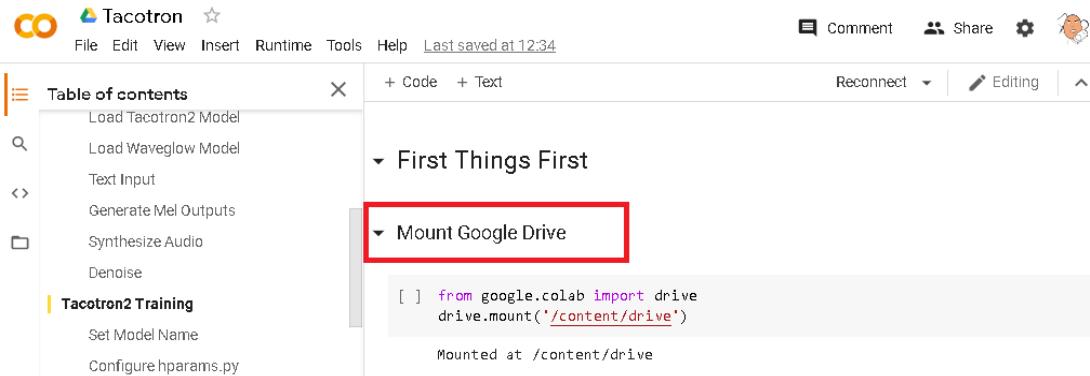
If Terminate is greyed out, just refresh the Colab tab in your browser then try again.

Important: We're provided free access to a physical GPU. If a GPU session is left idle it will terminate automatically and you will need to reconnect.

It's also important to note that sessions will only persist for a few hours, depending on the overall system usage of the Google Colab service.

Because sessions are temporary we'll need to connect our Colab session to our Google Drive storage so we can both access our dataset and save the checkpoints and log files from Tacotron and WaveGlow back to it.

First, a bit about Colab Notebooks. Colab Notebooks are made up of two types of cells. Text Cells and Code Cells.



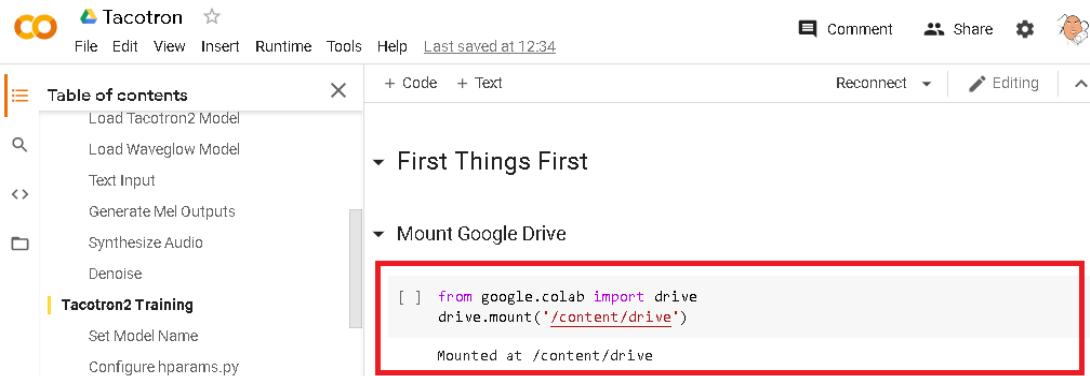
The screenshot shows a Google Colab interface with a 'Table of contents' sidebar on the left. The main area displays a 'Text' cell under the 'First Things First' section. The cell contains the following text:

```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

Below the code, it says 'Mounted at /content/drive'. A red box highlights the code block.

Figure 1352 - A text cell.

Text Cells are used to add labels.



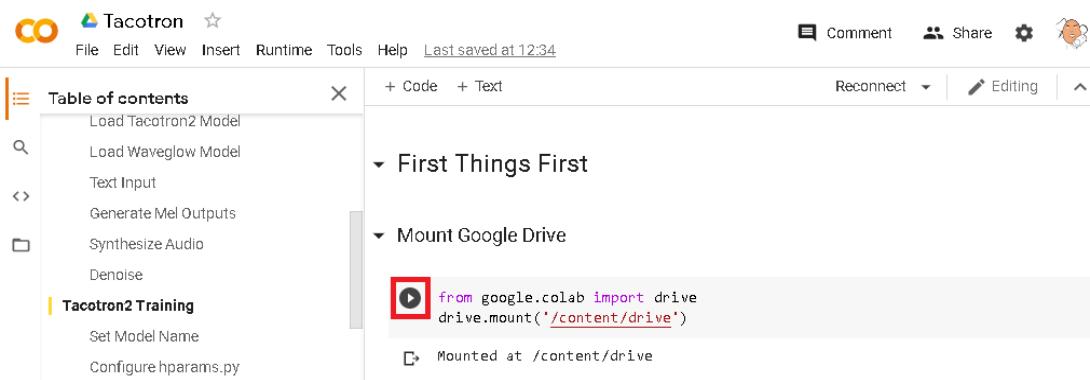
The screenshot shows a Google Colab interface with a 'Table of contents' sidebar on the left. The main area displays a 'Code' cell under the 'Mount Google Drive' section. The cell contains the following code:

```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

Below the code, it says 'Mounted at /content/drive'. A red box highlights the code block.

Figure 1353 - A code cell.

Code Cells are used to run commands or python code.



The screenshot shows a Google Colab interface with a 'Table of contents' sidebar on the left. The main area displays a 'Code' cell under the 'Mount Google Drive' section. The cell contains the following code:

```
[ ] from google.colab import drive
drive.mount('/content/drive')
```

Below the code, it says 'Mounted at /content/drive'. A red box highlights the play/run icon (a circular arrow) to the left of the cell.

Figure 1354 - Run cell button.

To execute a code cell, click on the **Run Cell** button.

Run the code cell under **Mount Google Drive**.

```
from google.colab import drive
drive.mount('/content/drive')

Go to this URL in a browser: https://accounts.google.com/o/oauth2/auth?client

Enter your authorization code:
```

Figure 1355 - Mount Google Drive.

We'll need to verify our connection before we can link our Google Drive storage to our Colab session. Click on the link produced in the cell.

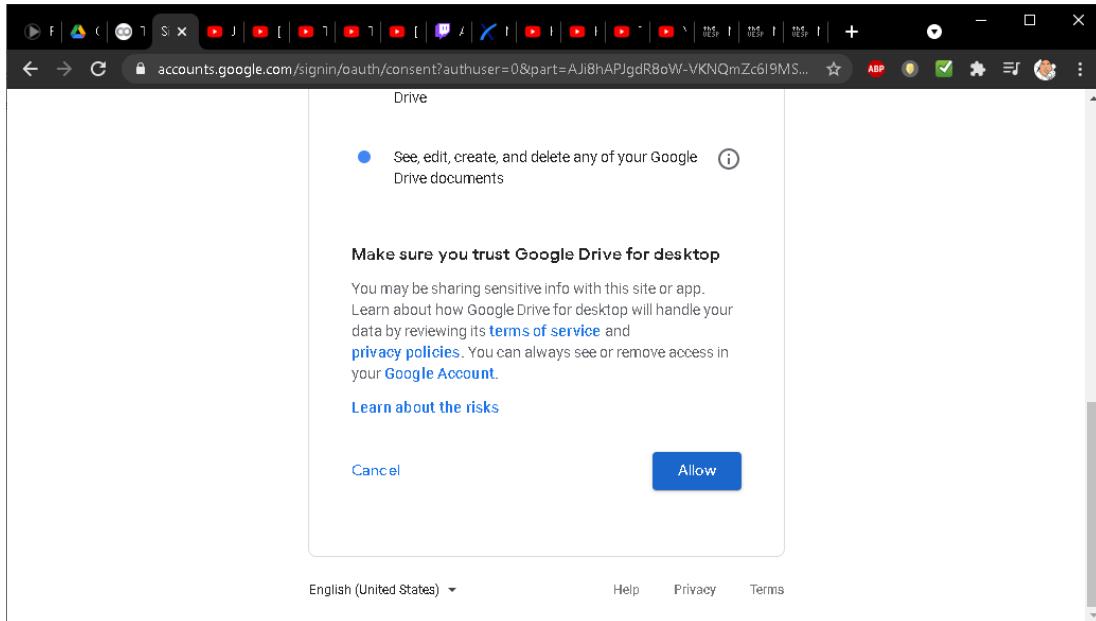


Figure 1356 - Google Drive verification.

Click Allow.

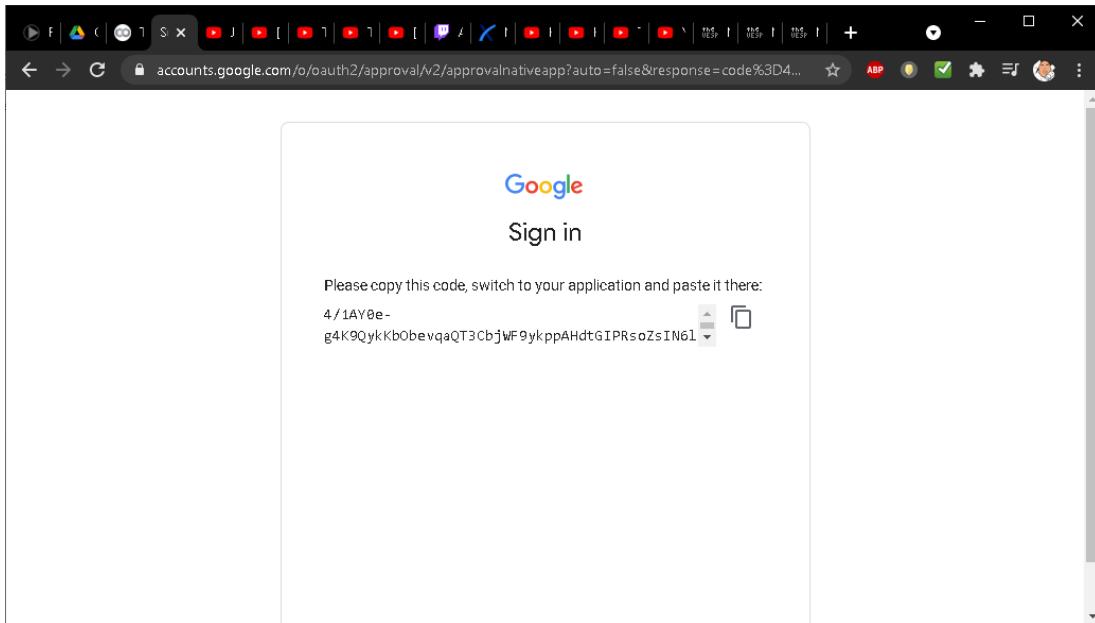


Figure 1357 - Authentication code.

Copy the authentication code and paste it into the text field back in the code cell like so, then press Enter.

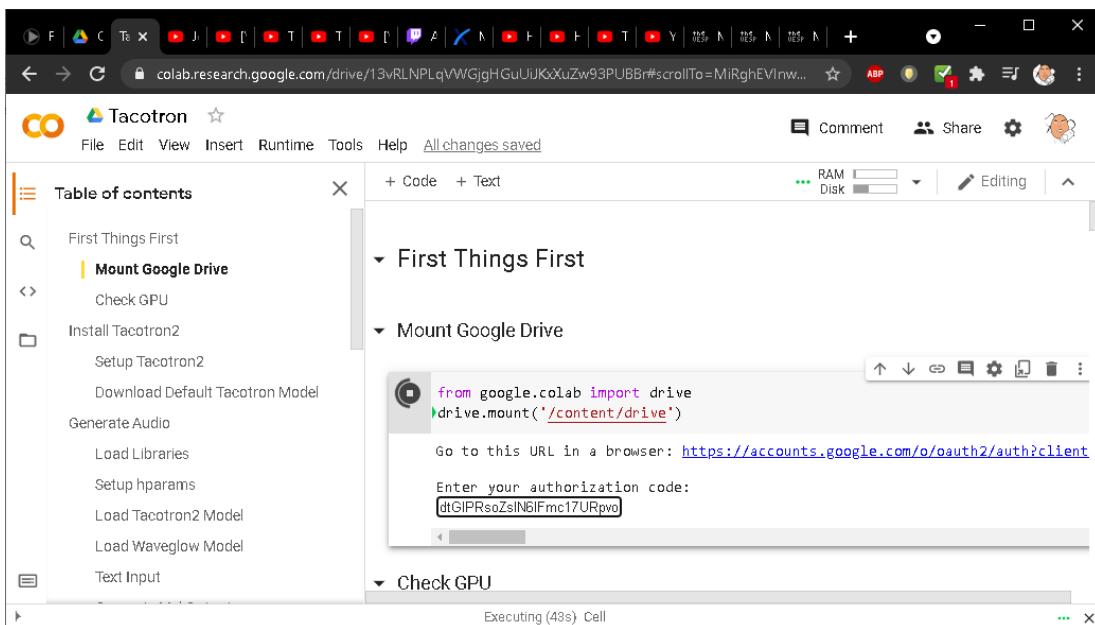
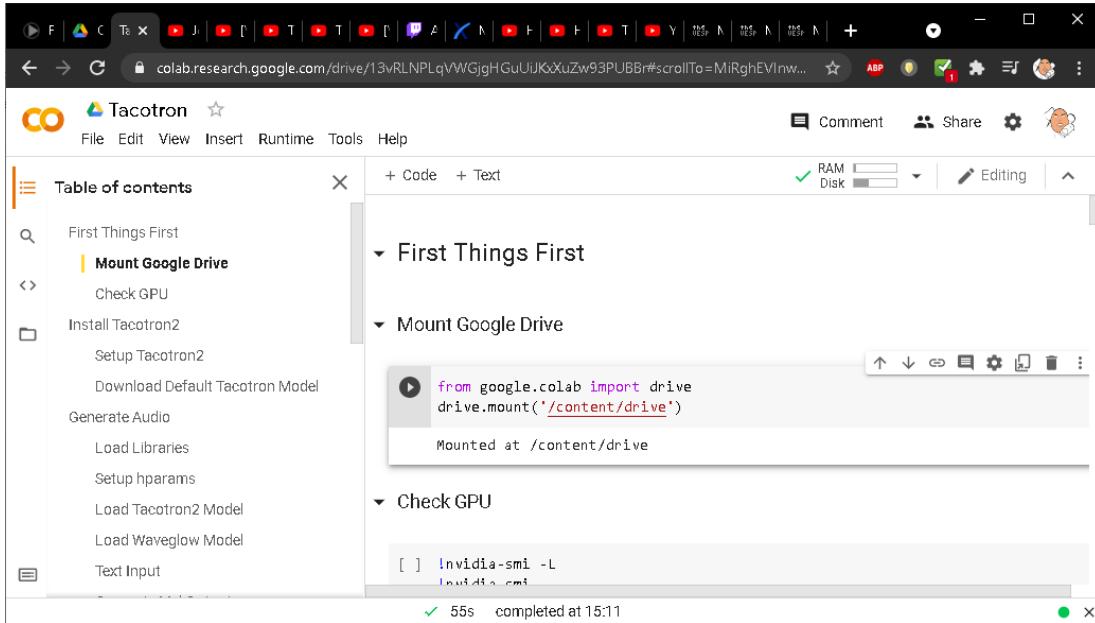


Figure 1358 - Entered the authentication code.



The screenshot shows a Google Colab notebook titled 'Tacotron'. The left sidebar contains a 'Table of contents' with sections like 'First Things First', 'Mount Google Drive', 'Check GPU', etc. The main area shows a code cell with the following content:

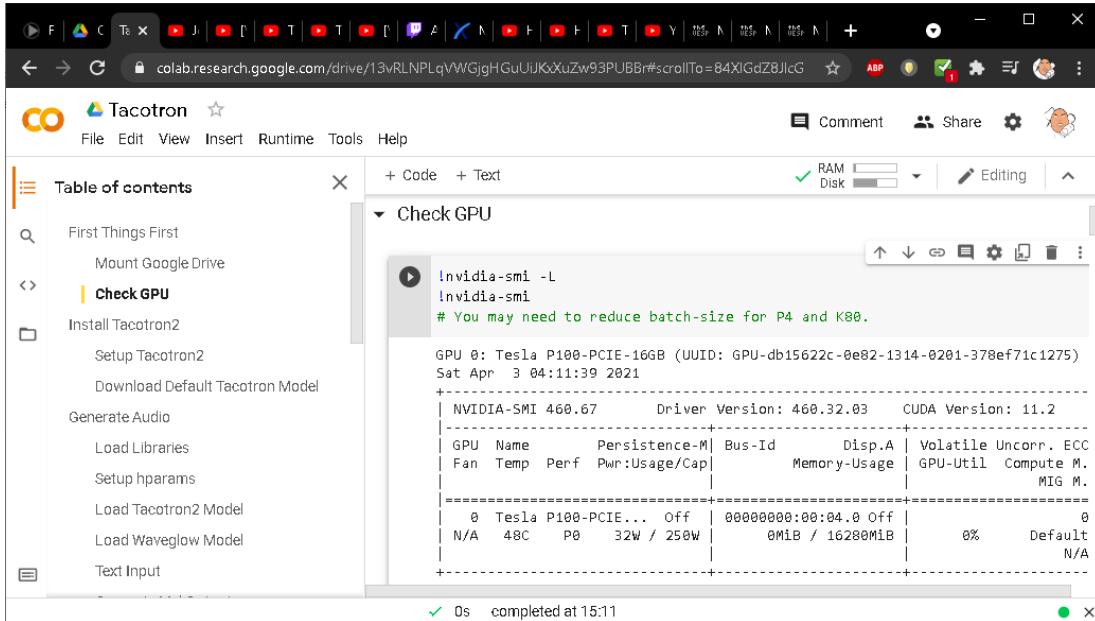
```
+ Code + Text
from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive
```

Below the code cell, there is a terminal output cell showing the command `!nvidia-smi -L` and its results.

Figure 1359 - Successfully mounted Google Drive storage.

Once it's mounted successfully, the cell will report 'Mounted at /content/drive'.

Run the code cell under **Check GPU**.



The screenshot shows a Google Colab notebook titled 'Tacotron'. The left sidebar contains a 'Table of contents' with sections like 'First Things First', 'Mount Google Drive', 'Check GPU', etc. The main area shows a code cell with the following content:

```
+ Code + Text
!nvidia-smi -L
!nvidia-smi
# You may need to reduce batch-size for P4 and K80.
```

Below the code cell, there is a terminal output cell showing the command `!nvidia-smi -L` and its results. The output includes information about GPU 0: Tesla P100-PCIE-16GB, including driver and CUDA versions, and a detailed table of GPU metrics.

GPU	Name	Persistence-M	Bus-Id	Disp.A	Volatile	Uncorr. ECC
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute M.
0	Tesla P100-PCIE...	Off	00000000:00:04.0	Off	0%	Default
N/A	48C	P0	32W / 250W	0MiB / 16280MiB		N/A

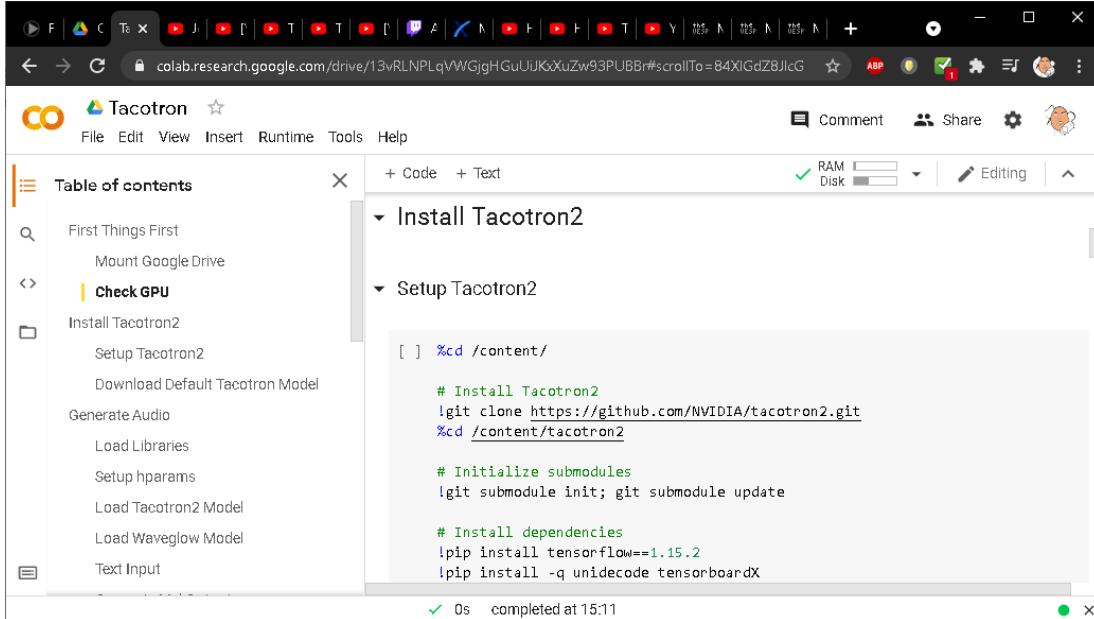
Figure 1360 - Check GPU.

This will tell us what kind of GPU has been assigned to our session. For P4 and K80 GPUs you may need to lower the batch_size settings later on if you run into an 'Out of Memory' error when you start training.

TRAINING A TACOTRON MODEL IN GOOGLE COLAB

Scroll down to the section labelled **Install Tacotron 2**.

Run the code cell under **Setup Tacotron 2**. This will install and Tacotron 2 and its dependencies to your Colab session.



The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar contains a "Table of contents" with sections like "First Things First", "Check GPU", and "Install Tacotron2". The main area shows a code cell under the "Install Tacotron2" section:

```
[ ] %cd /content/
# Install Tacotron2
!git clone https://github.com/NVIDIA/tacotron2.git
%cd tacotron2

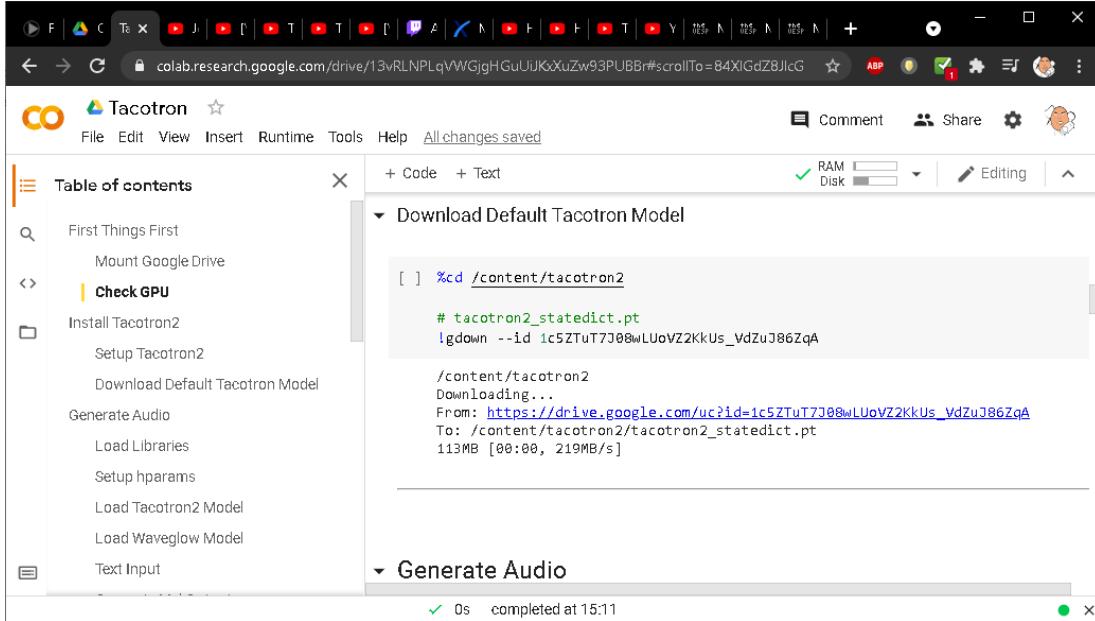
# Initialize submodules
!git submodule init; git submodule update

# Install dependencies
!pip install tensorflow==1.15.2
!pip install -q unidecode tensorboardX
```

The code cell has a green checkmark icon and the status "0s completed at 15:11".

Figure 1361 - Installing Tacotron 2.

Run the code cell under **Download Default Tacotron Model**. This will download the default Tacotron model that we'll be using to warm-start our own model from.



The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar contains a "Table of contents" with sections like "First Things First", "Mount Google Drive", "Check GPU", "Install Tacotron2", "Setup Tacotron2", "Download Default Tacotron Model", "Generate Audio", "Load Libraries", "Setup hparams", "Load Tacotron2 Model", "Load WaveGlow Model", and "Text Input". The main area shows a code cell under the "Download Default Tacotron Model" section:

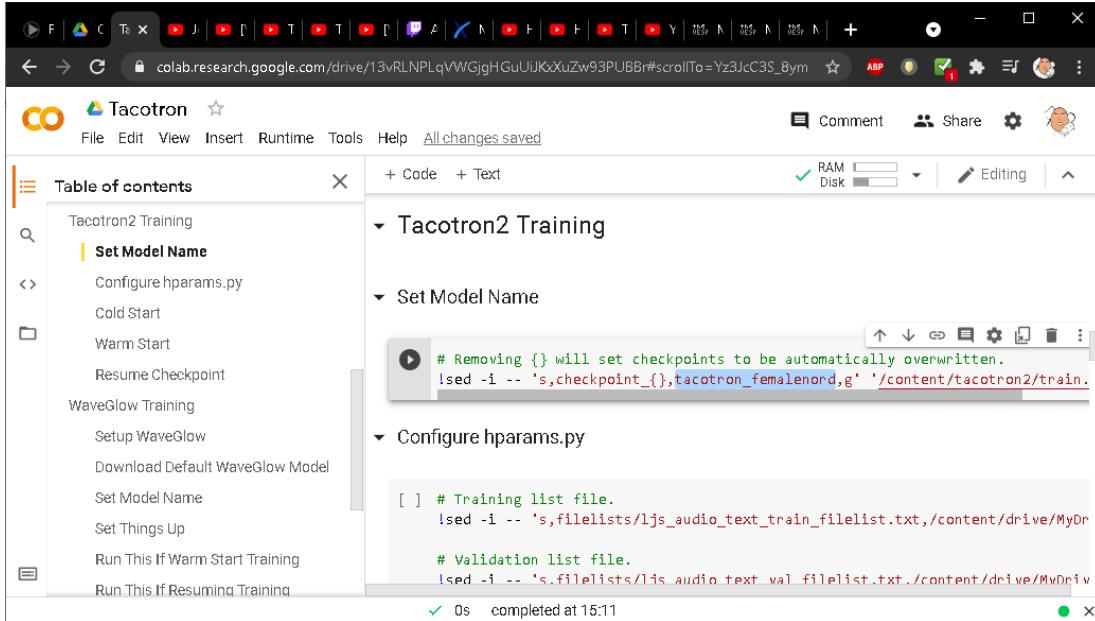
```
[ ] %cd /content/tacotron2
# tacotron2_statedict.pt
!gdown --id 1c52TuT7J08wLUoVZ2KkUs_VdzuJ86ZqA
/content/tacotron2
Downloading...
From: https://drive.google.com/uc?id=1c52TuT7J08wLUoVZ2KkUs\_VdzuJ86ZqA
To: /content/tacotron2/tacotron2_statedict.pt
113MB [00:00, 219MB/s]
```

The status bar at the bottom indicates "0s completed at 15:11".

Figure 1362 - Download default Tacotron model.

Scroll down to Tacotron 2 Training.

In the code cell beneath **Set Model Name**, change 'tacotron_femalenord' to match the voice type of the dataset you've prepared.



The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar contains a "Table of contents" with sections like "Tacotron2 Training", "Set Model Name", "Configure hparams.py", "Cold Start", "Warm Start", "Resume Checkpoint", "WaveGlow Training", "Setup WaveGlow", "Download Default WaveGlow Model", "Set Model Name", "Set Things Up", "Run This If Warm Start Training", and "Run This If Resuming Training". The main area shows a code cell under the "Set Model Name" section:

```
# Removing {} will set checkpoints to be automatically overwritten.
!sed -i -- 's,checkpoint_{},tacotron_femalenord,g' '/content/tacotron2/train.'
```

The status bar at the bottom indicates "0s completed at 15:11".

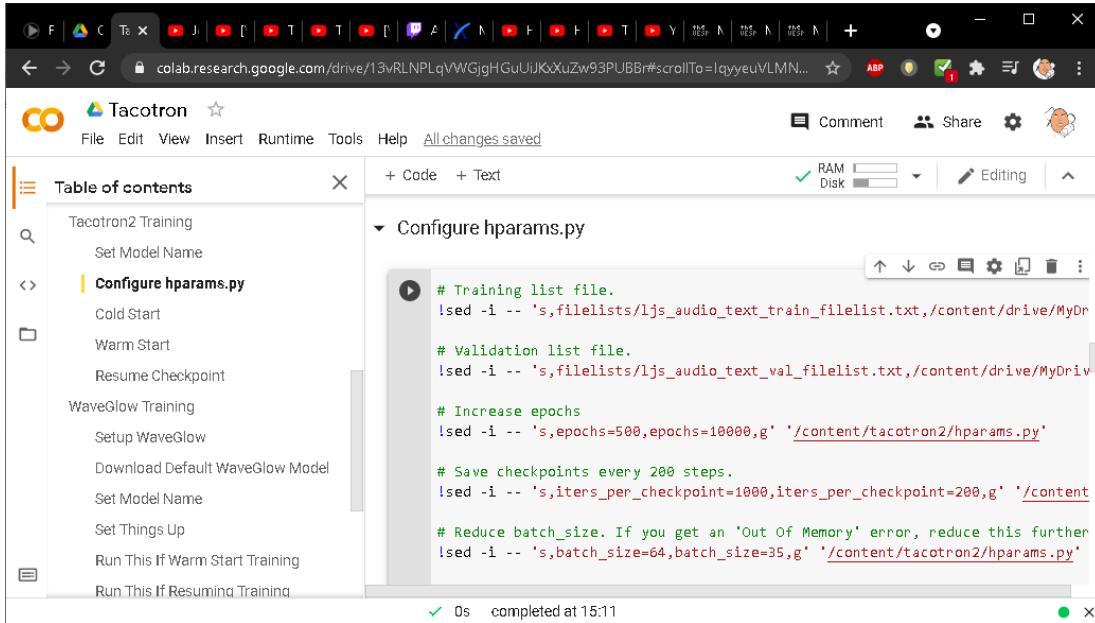
Figure 1363 - Set model name.

By default Tacotron saves individual checkpoints as it trains as ‘checkpoint_{}’. Removing ‘{}’ from the model name tells Tacotron to just keep one checkpoint and overwrite it when it needs to save a new one.

Google Drive is limited to 15GB of storage space and this can be used up pretty quickly when training a Tacotron or WaveGlow model if we save checkpoints individually.

Click Run Cell once you’ve changed the model name.

Scroll down to **Configure hparams.py**.



The screenshot shows a Google Colab notebook titled 'Tacotron'. The left sidebar contains a 'Table of contents' with sections like 'Tacotron2 Training', 'Configure hparams.py', 'Cold Start', etc. The main area shows a code cell with the following content:

```
# Training list file.
!sed -i -- '$,filelists/ljs_audio_text_train_filelist.txt,/content/drive/MyDrive/filelists/femalenord_training.txt' ./content/tacotron2/hparams.py

# Validation list file.
!sed -i -- '$,filelists/ljs_audio_text_val_filelist.txt,/content/drive/MyDrive/filelists/femalenord_validation.txt' ./content/tacotron2/hparams.py

# Increase epochs
!sed -i -- '$,epochs=500,epochs=1000,g' './content/tacotron2/hparams.py'

# Save checkpoints every 200 steps.
!sed -i -- '$,iters_per_checkpoint=1000,iters_per_checkpoint=200,g' './content/tacotron2/hparams.py'

# Reduce batch_size. If you get an 'Out Of Memory' error, reduce this further
!sed -i -- '$,batch_size=64,batch_size=35,g' './content/tacotron2/hparams.py'
```

The status bar at the bottom indicates '0s completed at 15:11'.

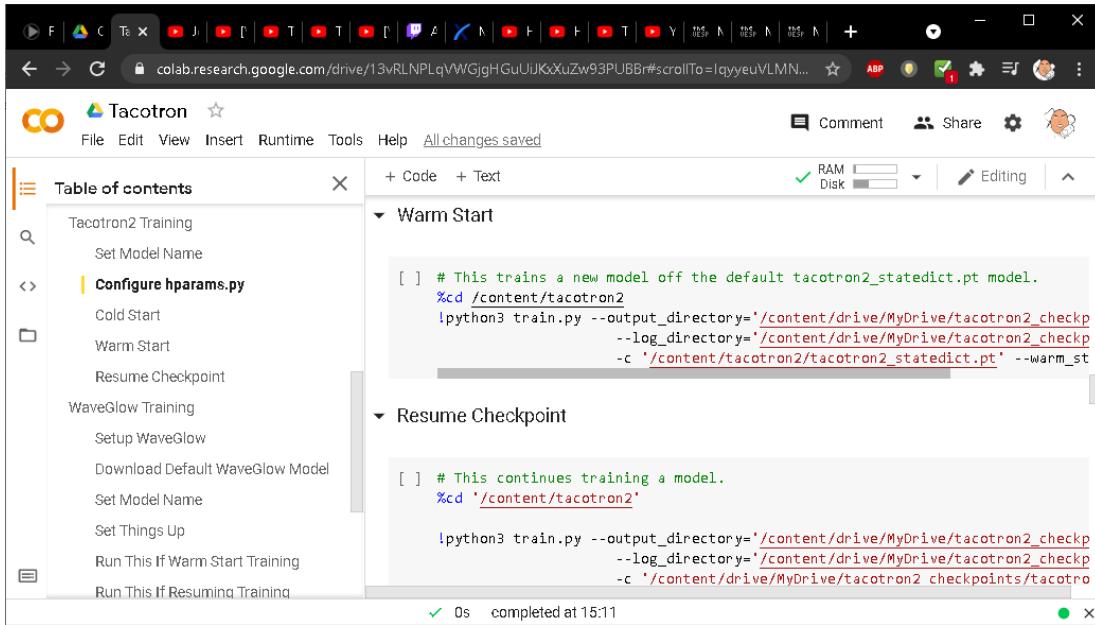
Figure 1364 - Configure hparams.py.

Set the path to your training and validation text files. By default, I have the training file path set to ‘/content/drive/MyDrive/filelists/femalenord_training.txt’ and the validation file path set to ‘/content/drive/MyDrive/filelists/femalenord_validation.txt’.

The other settings can be left as is for now.

Click Run Cell once you’ve changed the file names for the training and validation files.

Scroll down to **Warm Start**.



The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar contains a "Table of contents" with sections like "Tacotron2 Training", "Configure hparams.py", "Cold Start", "Warm Start", "Resume Checkpoint", "WaveGlow Training", and "Run This If Resuming Training". The main area shows code snippets under the "Warm Start" section:

```
[ ] # This trains a new model off the default tacotron2_statedict.pt model.
!python3 train.py --output_directory='/content/drive/MyDrive/tacotron2_checkpoints'
--log_directory='/content/drive/MyDrive/tacotron2_checkpoints'
-c '/content/tacotron2/tacotron2_statedict.pt' --warm_start
```

Below this, another snippet is shown under "Resume Checkpoint":

```
[ ] # This continues training a model.
!cd '/content/tacotron2'
!python3 train.py --output_directory='/content/drive/MyDrive/tacotron2_checkpoints'
--log_directory='/content/drive/MyDrive/tacotron2_checkpoints'
-c '/content/drive/MyDrive/tacotron2_checkpoints/tacotron2_statedict.pt'
```

Figure 1365 - Warm Start.

Warm Start allows us to begin training a new model off the default Tacotron model. This is useful for small datasets like the ones we make from Skyrim or Oblivion.

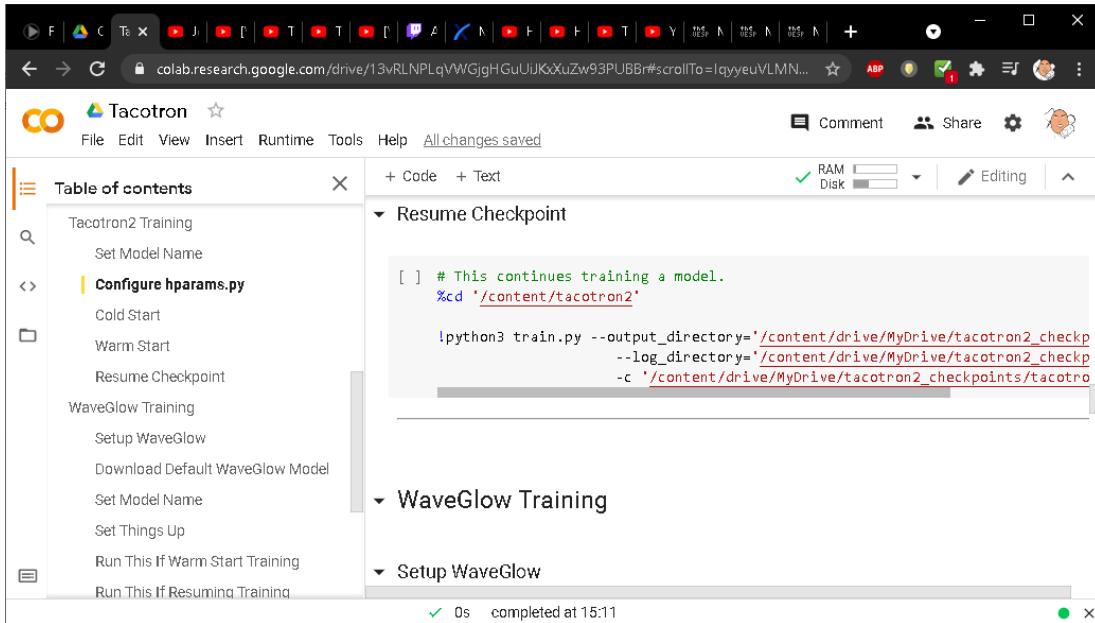
Models will be saved to a new folder in Google Drive called 'tacotron2_checkpoints'. The log directory will be saved there as well.

Click Run Cell to begin training a new model.

Assuming you don't run into any errors, you should see a new model in the 'tacotron2_checkpoints' folder after 200 iterations. Every 200 iterations your progress will be saved, overwriting the existing checkpoint.

To resume training an existing model, scroll down to **Resume Checkpoint**.

Note: You'll need to make sure you've run **Setup Tacotron2**, **Set Model Name** and **Configure hparams.py** first before resuming.



```
[ ] # This continues training a model.
%cd '/content/tacotron2'

!python3 train.py --output_directory='/content/drive/MyDrive/tacotron2_checkpoints'
--log_directory='/content/drive/MyDrive/tacotron2_checkpoints'
-c '/content/drive/MyDrive/tacotron2_checkpoints/tacotron_femalenord'
```

Figure 1366 - Resume Checkpoint.

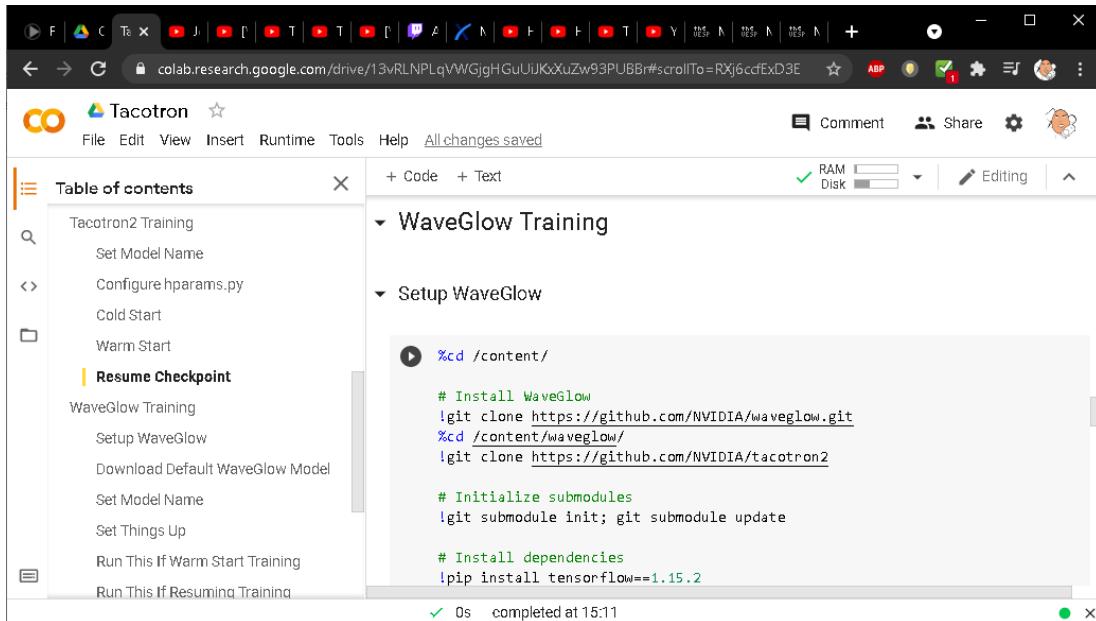
For the `-c` switch, change the name of the model and ensure the path to it is correct. By default I have this switch set to `'/content/drive/MyDrive/tacotron2_checkpoints/tacotron_femalenord'`.

Click Run Cell to resume training a model.

TRAINING A WAVEGLOW MODEL IN GOOGLE COLAB

Scroll down to the section labelled **Install WaveGlow**.

Run the code cell under **Setup WaveGlow**.



The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar contains a "Table of contents" with sections like "Tacotron2 Training", "Configure hparams.py", "Cold Start", "Warm Start", and "Resume Checkpoint". The main area shows a code cell under the "Setup WaveGlow" section:

```
%cd /content/
# Install WaveGlow
!git clone https://github.com/NVIDIA/waveglow.git
%cd /content/waveglow/
!git clone https://github.com/NVIDIA/tacotron2

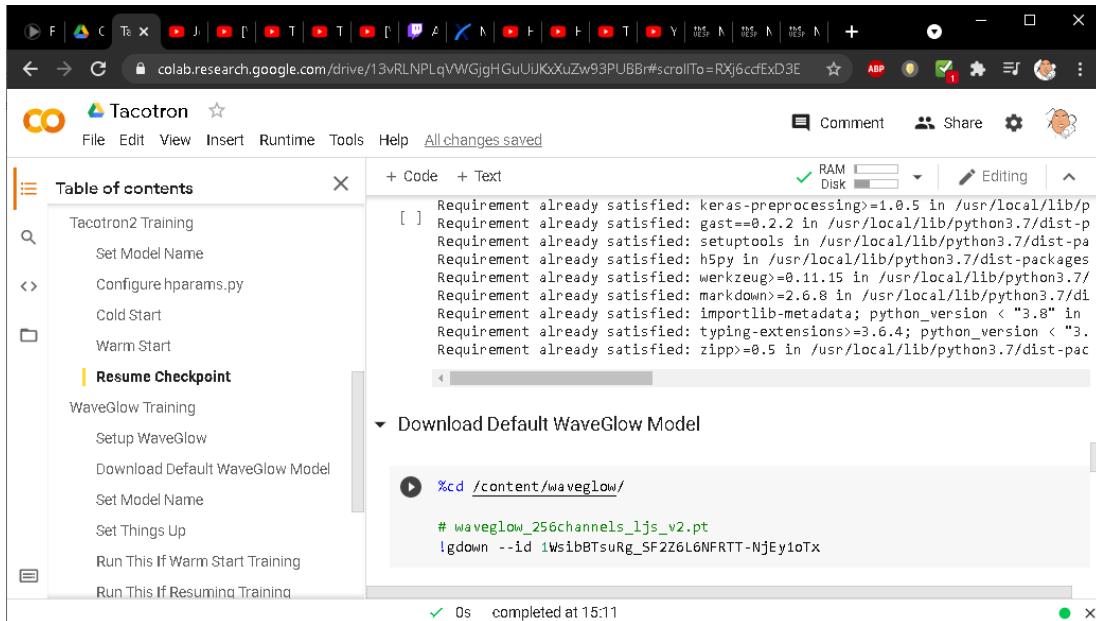
# Initialize submodules
!git submodule init; git submodule update

# Install dependencies
!pip install tensorflow==1.15.2
```

The code cell has a green checkmark icon and the status "0s completed at 15:11".

Figure 1367 - Setup WaveGlow.

Run the code cell under **Download Default WaveGlow Model**. This will download the default WaveGlow model that we'll be using to warm-start our own model from.



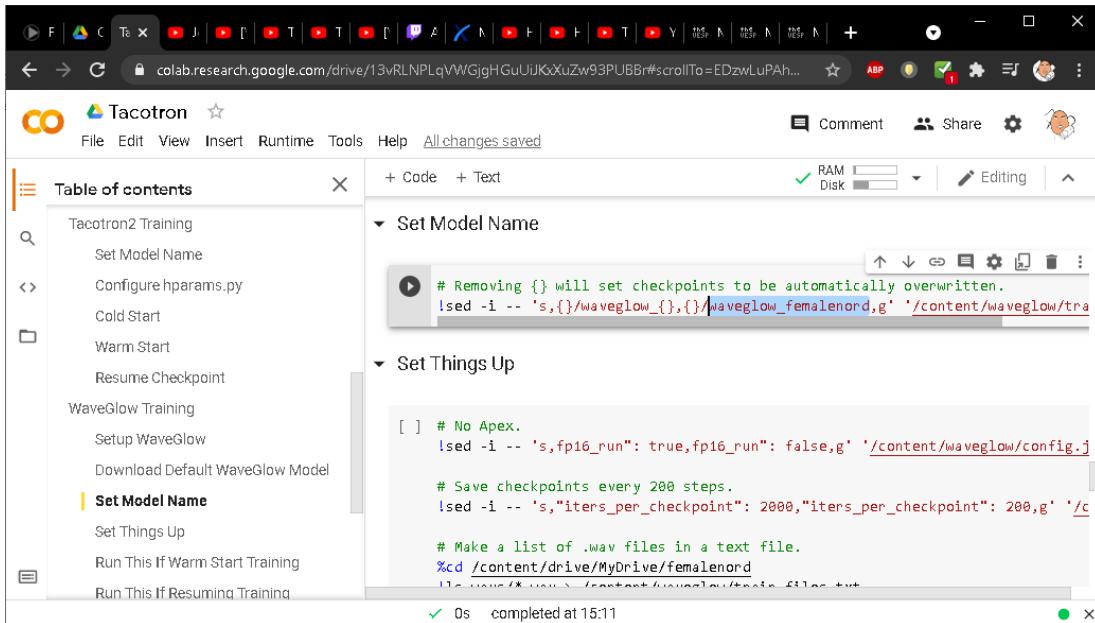
The screenshot shows the same Google Colab notebook. The "Table of contents" sidebar is visible. The main area shows a code cell under the "Download Default WaveGlow Model" section:

```
%cd /content/waveglow/
# waveglow_256channels_ljs_v2.pt
!gdown --id 1WsibBTsRg_SFZ6L6NFRRT-NjEyioTx
```

The code cell has a green checkmark icon and the status "0s completed at 15:11".

Figure 1368 - Download Default WaveGlow Model.

In the code cell beneath **Set Model Name**, change ‘waveglow_femalenord’ to match the voice type of the dataset you’ve prepared.



```
# Removing {} will set checkpoints to be automatically overwritten.
!sed -i -- 's,{}/waveglow_{},{}/waveglow_femalenord,g' '/content/waveglow/training/config.py'

# No Apex.
!sed -i -- 's,fp16_run": true,fp16_run": false,g' '/content/waveglow/config.json'

# Save checkpoints every 200 steps.
!sed -i -- 's,"iters_per_checkpoint": 2000,"iters_per_checkpoint": 200,g' '/content/waveglow/config.json'

# Make a list of .wav files in a text file.
!cd /content/drive/MyDrive/femalenord
!ls *.wav > /content/drive/MyDrive/femalenord/tracklist.txt
```

Figure 1369 - Set Model Name.

By default WaveGlow saves individual checkpoints as it trains as ‘waveglow_{}’. Removing ‘{}’ from the model name tells WaveGlow to just keep one checkpoint and overwrite it when it needs to save a new one.

Note: Google Drive is limited to 15 GB of storage space and this can be used up pretty quickly when training a Tacotron or WaveGlow model if we save checkpoints individually.

Click Run Cell once you’ve changed the model name.

Scroll down to **Set Things Up**.

```

# No Apex.
!sed -i -- 's,fp16_run": true,fp16_run": false,g' '/content/waveglow/config.json'

# Save checkpoints every 200 steps.
!sed -i -- 's,"iters_per_checkpoint": 2000,"iters_per_checkpoint": 200,g' '/content/waveglow/config.json'

# Make a list of .wav files in a text file.
%cd /content/drive/MyDrive/femalenord
!ls Wavs/*.wav > /content/waveglow/train_files.txt

# Save log files for Tensorboard.
!sed -i -- 's,"with_tensorboard": false,"with_tensorboard": true,g' '/content/waveglow/config.json'

# Make sure the text files contain the full paths to the .wav files.
%cd /content/waveglow/
!sed -i -- 's,wavs/,/content/drive/MyDrive/femalenord/wavs/,g' '/content/waveglow/train_files.txt'

```

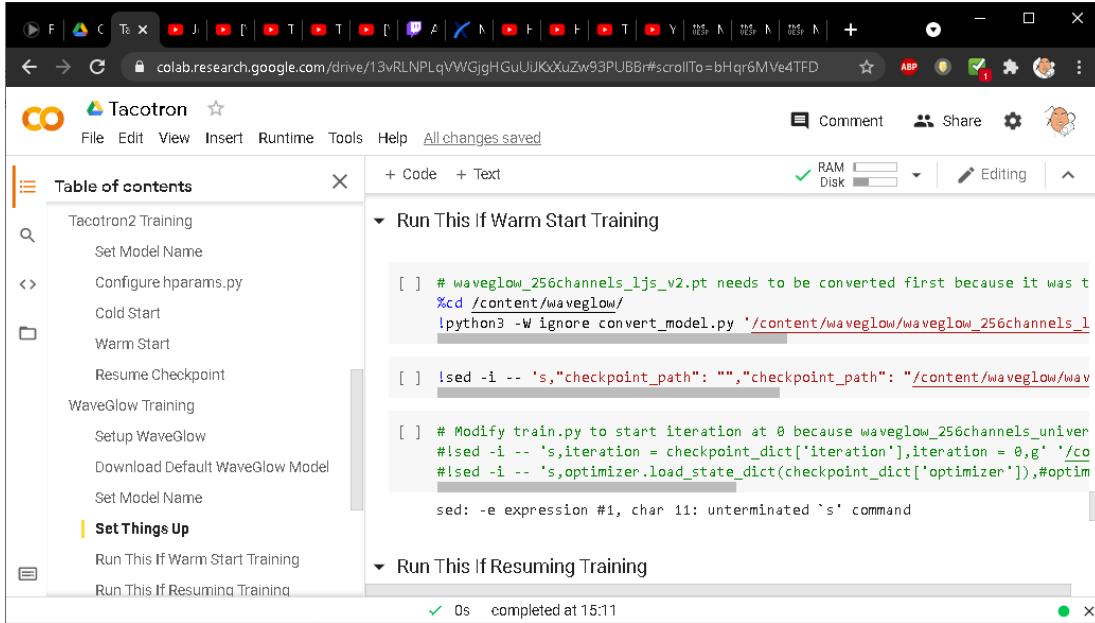
Figure 1370 - Set Things Up.

Set the path to the folder containing your .wav files in the areas highlighted in the screenshot above.

Click Run Cell once you've changed those paths.

We won't need the training and validation files we made earlier for Tacotron. This cell will create its own train_files.txt file for WaveGlow to use. It will also tell WaveGlow to save checkpoints to the 'waveglow_checkpoints' folder back in Google Drive. The log directory will be saved there as well.

If you're training a new model, run the first two code cells under **Run This If Warm Starting**.



The screenshot shows a Google Colab notebook titled "Tacotron". On the left, there's a "Table of contents" sidebar with sections like "Tacotron2 Training", "Cold Start", "Warm Start", and "WaveGlow Training". The main area displays two code cells under the heading "Run This If Warm Start Training". The first cell contains:

```
[ ] # waveglow_256channels_ljs_v2.pt needs to be converted first because it was t
%cd /content/waveglow/
!python3 -W ignore convert_model.py '/content/waveglow/waveglow_256channels_l
```

The second cell contains:

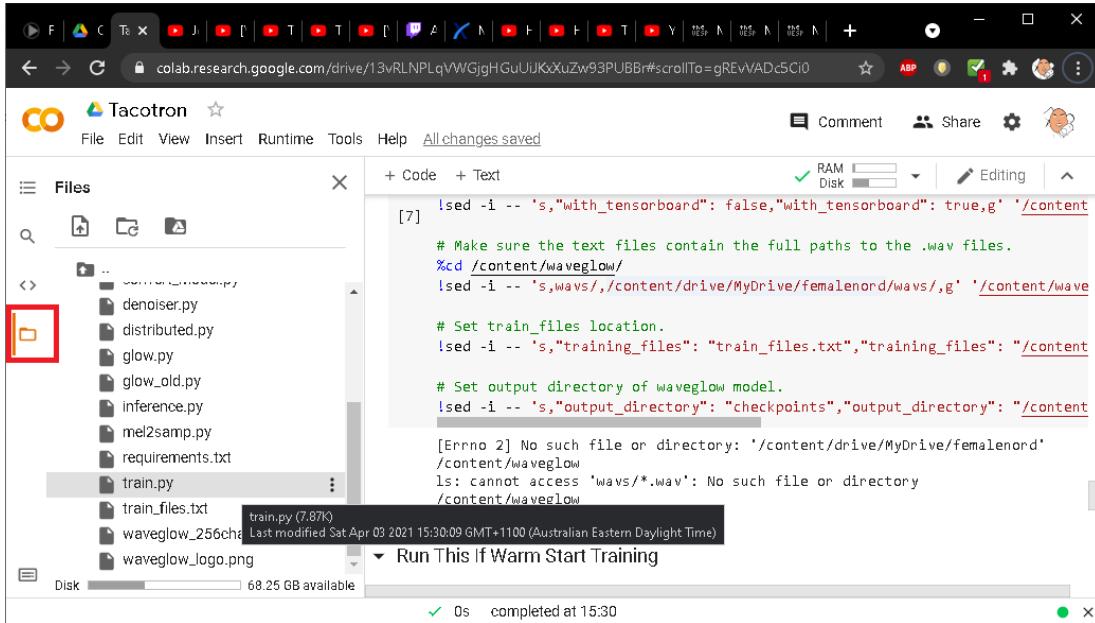
```
[ ] !sed -i -- 's,"checkpoint_path": "", "checkpoint_path": "/content/waveglow/wav
```

Below these, another section titled "Run This If Resuming Training" is partially visible.

Figure 1371 - Run This If Warm Starting.

This will convert the default WaveGlow model 'waveglow_256channels_ljs_v2.pt' to allow us to warm start from it.

Click on Browse on the left hand side. Navigate to content > waveglow and open train.py.



The screenshot shows a Google Colab notebook titled "Tacotron". On the left, there's a "Files" sidebar with a red box highlighting the folder icon. Inside the folder, files like "denoiser.py", "distributed.py", "glow.py", "glow_old.py", "inference.py", "mel2samp.py", "requirements.txt", "train.py", and "train_files.txt" are listed. The main area shows the content of "train.py". A red box highlights the file name "train.py (7.87K)". The code in the editor is:

```
[7] !sed -i -- 's,"with_tensorboard": false,"with_tensorboard": true,g' '/content
```

Below the code, an error message is shown:

```
# Make sure the text files contain the full paths to the .wav files.
%cd /content/waveglow/
!sed -i -- 's,wavs,/content/drive/MyDrive/femalenord/wavs,/g' '/content/wave

# Set train_files location.
!sed -i -- 's,"train_files": "train_files.txt","training_files": "/content
```

Further down:

```
# Set output directory of waveglow model.
!sed -i -- 's,"output_directory": "checkpoints","output_directory": "/content
```

An error message follows:

```
[Errno 2] No such file or directory: '/content/drive/MyDrive/femalenord'
/content/waveglow
ls: cannot access 'wavs/*.wav': No such file or directory
/content/waveglow
```

Figure 1372 - Opening train.py.

Scroll down to line 44 in train.py. Change

```
iteration = checkpoint_dict['iteration']
```

to

```
iteration = 0
```

Comment out line 45 by adding a '#' in front.

```

  + Code + Text
  [7] !sed -i -- '$, "with_tensorboard": false, "with'
  # Make sure the text files contain the full p
  %cd /content/waveglow/
  !sed -i -- '$, "wavs": "/content/drive/MyDrive/fe
  # Set train_files location.
  !sed -i -- '$, "training_files": "train_files.
  # Set output directory of waveglow model.
  !sed -i -- '$, "output_directory": "checkpoint
[Errno 2] No such file or directory: '/content/waveglow
ls: cannot access 'wavs/*.wav': No such file
  /content/waveglow
  iteration = 0
#optimizer.load_state_dict(
  assert os.path.isfile(ch
  checkpoint_dict = torch.
  iteration = 0
#optimizer.load_state_dict(
  model_for_loading = ched
  model.load_state_dict(mo
  print("Loaded checkpoint
  checkpoint_path, i
  return model, optimizer,
def save_checkpoint(model, o

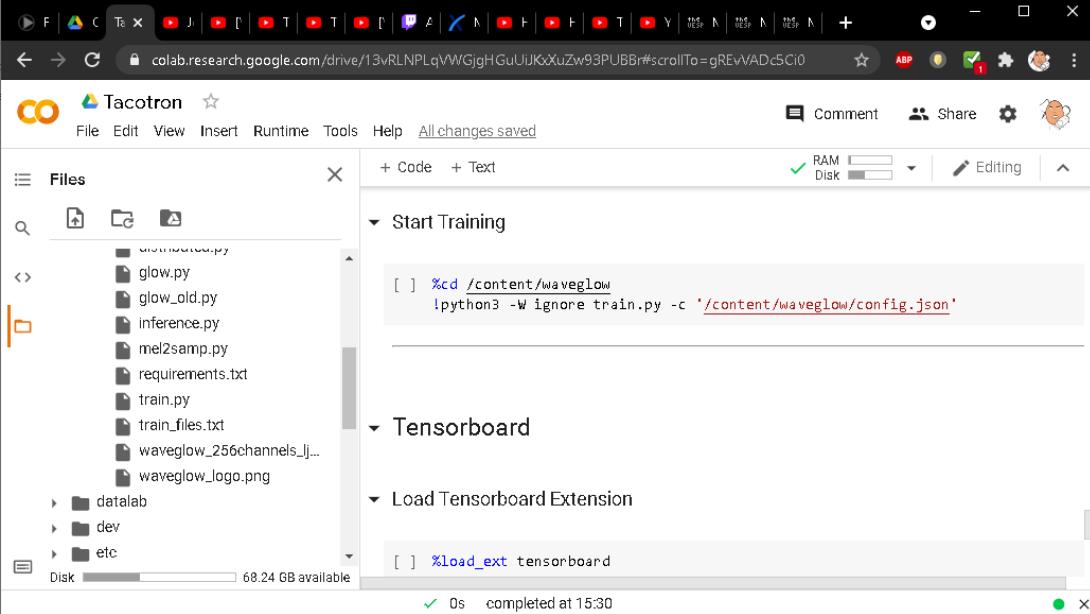
```

Figure 1373 - train.py modified.

Press CTRL+S to save the document, then close train.py.

This will allow us to use the default WaveGlow model we just converted.

Scroll down to **Start Training**.



The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar displays a file tree with files like "distributions.py", "glow.py", "glow_old.py", "inference.py", "mel2samp.py", "requirements.txt", "train.py", "train_files.txt", "waveglow_256channels_lj...", and "waveglow_logo.png". The main area has two code cells. The top cell, under the heading "Start Training", contains the command: `[] %cd /content/waveglow
!python3 -W ignore train.py -c '/content/waveglow/config.json'`. The bottom cell, under the heading "Tensorboard", contains the command: `[] %load_ext tensorboard`. A progress bar at the bottom indicates "0s completed at 15:30".

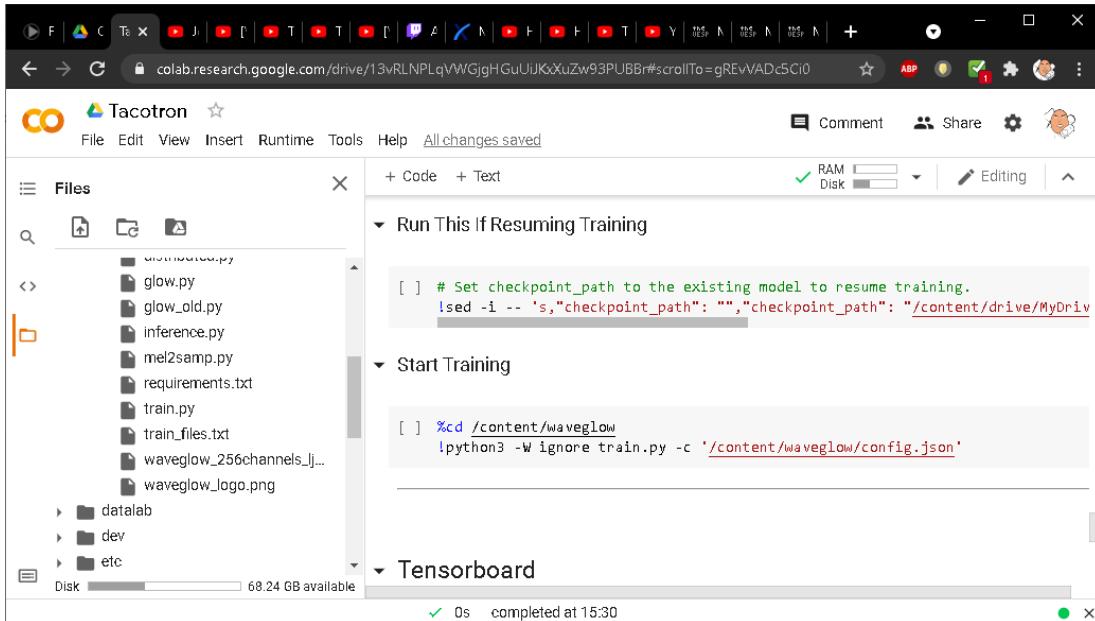
Figure 1374 - Start Training.

Run the cell below it to begin training a new WaveGlow model.

Assuming you don't run into any errors, you should see a new model in the 'waveglow_checkpoints' folder after 200 iterations. Every 200 iterations your progress will be saved, overwriting the existing checkpoint.

To resume training an existing model, scroll down to **Run This If Resuming Training**.

Note: you'll need to make sure you've run **Setup WaveGlow**, **Set Model Name**, and **Set Things Up** first.



The screenshot shows a Google Colab notebook interface. The left sidebar displays a file tree for a project named 'Tacotron'. The main area contains three code cells:

- Run This If Resuming Training:**

```
[ ] # Set checkpoint_path to the existing model to resume training.
!sed -i -- 's,"checkpoint_path": "", "checkpoint_path": "/content/drive/MyDrive/waveglow_femalenord"' ./train.py
```
- Start Training:**

```
[ ] %cd /content/waveglow
!python3 -W ignore train.py -c '/content/waveglow/config.json'
```
- Tensorboard:**

```
[ ] tensorboard --logdir=/content/tensorboard --port=6006
```

The notebook status bar at the bottom indicates '0s completed at 15:30'.

Figure 1375 - Run This If Resuming Training.

By default this cell is set to resume training the waveglow_femalenord model, so change this to match the file name of the model you're training, then click Run Cell.

Now scroll down to **Start Training** and click Run Cell to resume training a model.

Training will continue from the last saved checkpoint.

CHECKING PROGRESS IN TENSORBOARD

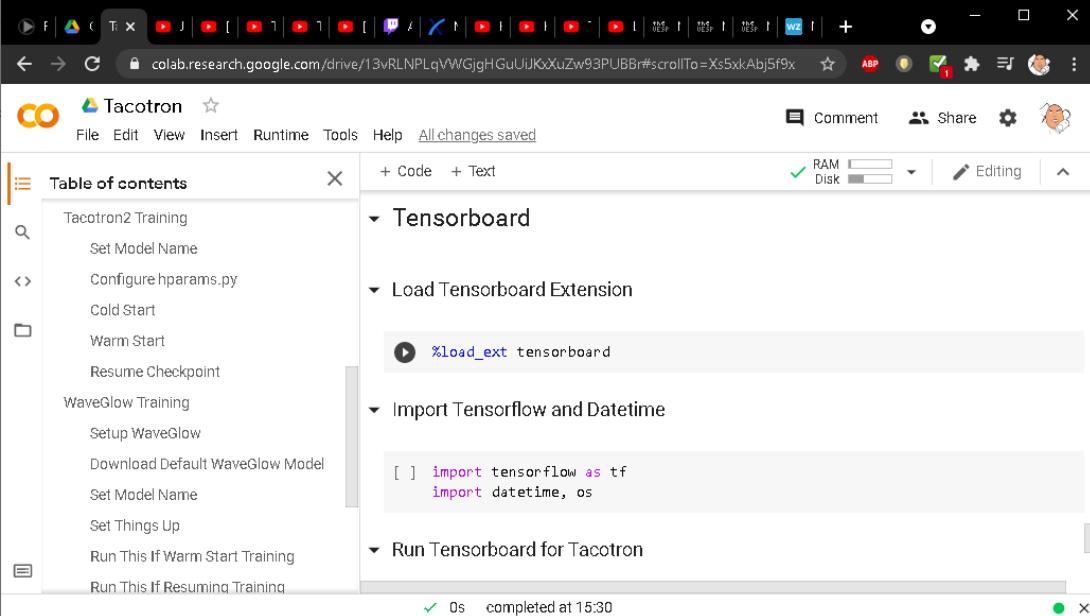
We can check the progress of our Tacotron and WaveGlow model in Tensorboard.

Tensorboard will read the log files from the logdir folder that was saved to our checkpoint directories. In our example we saved our tacotron checkpoints to ‘tacotron_checkpoints’. For WaveGlow, that folder is ‘waveglow_checkpoints’.

I would recommend testing your models after every 10,000 iterations just to check that things are on the right track. See the next section titled **Synthesizing audio using the models we’ve trained** for the steps on how to generate audio to do this.

Generally speaking, training is considered ‘done’ once we’re satisfied with the output.

Scroll down to **Tensorboard** and run the code cell under **Load Tensorboard Extension**.

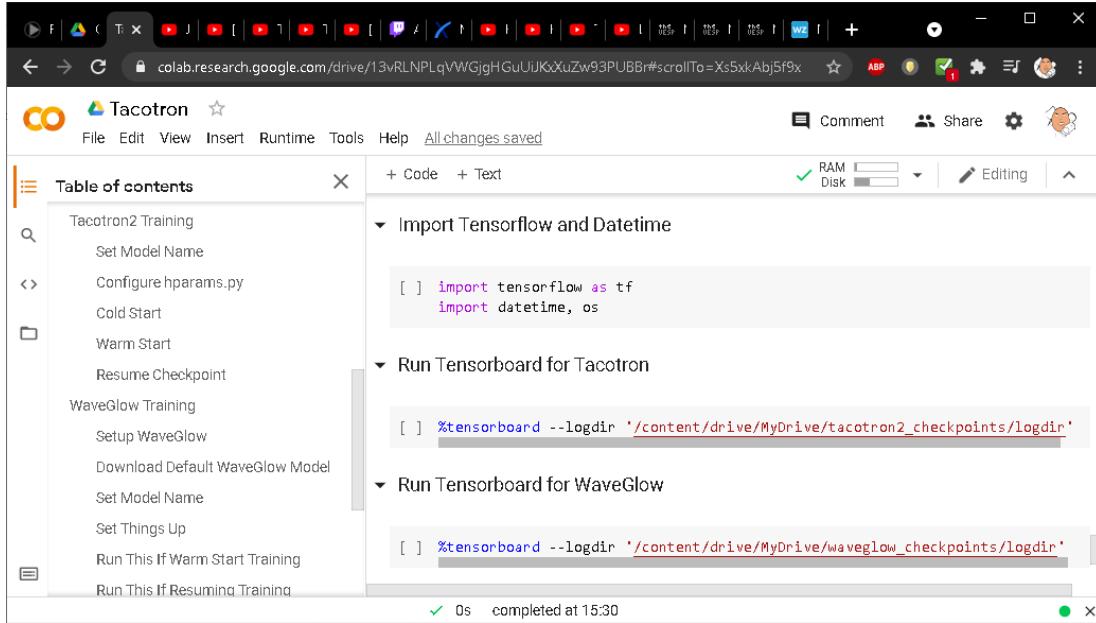


The screenshot shows a Jupyter Notebook interface on Google Colab. The left sidebar has a 'Table of contents' section with various training-related entries like 'Tacotron2 Training', 'WaveGlow Training', etc. The main area shows a code cell under the 'Load Tensorboard Extension' section:

```
[ ] %load_ext tensorboard
```

Figure 1376 - Load Tensorboard Extension.

Next, run the code cell under **Import Tensorflow and Datetime**.

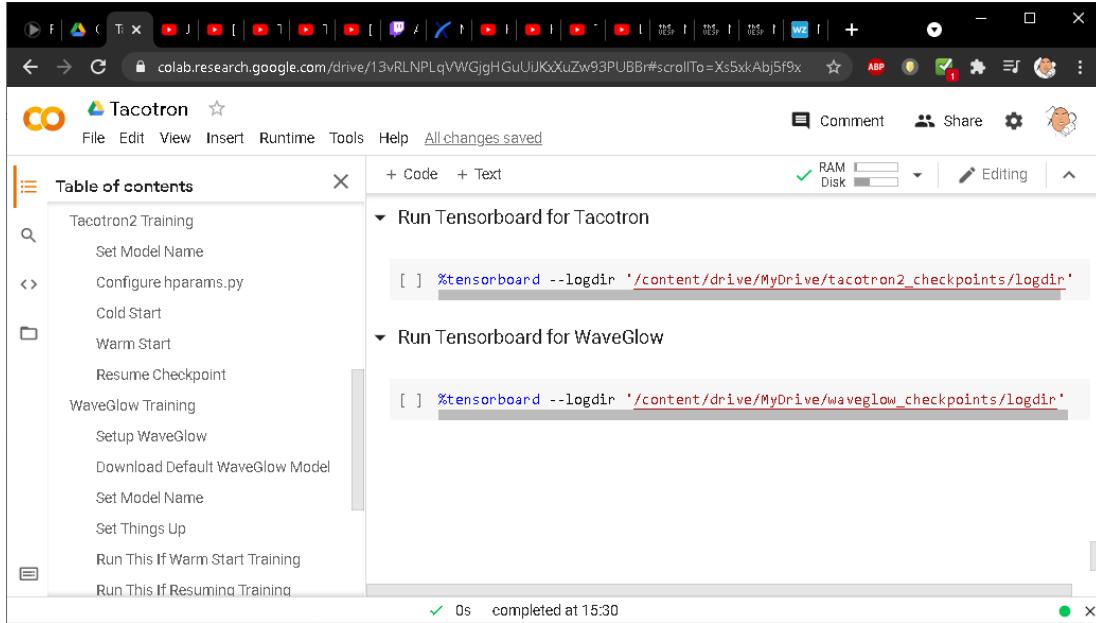


The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar contains a "Table of contents" with various training and setup options. The main area displays a code cell under the heading "Import Tensorflow and Datetime". The code in the cell is:

```
[ ] import tensorflow as tf
import datetime, os
```

Figure 1377 - Import Tensorflow and Datetime.

Run either the code cell under **Run Tensorboard for Tacotron** or **Run Tensorboard for WaveGlow** depending on which one you want to check.



The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar contains a "Table of contents" with various training and setup options. The main area displays two code cells under the headings "Run Tensorboard for Tacotron" and "Run Tensorboard for WaveGlow". Both cells contain the same command:

```
[ ] %tensorboard --logdir '/content/drive/MyDrive/tacotron2_checkpoints/logdir'
```

Figure 1378 - Run Tensorboard.

In the Scalars section, the graphs we need to pay attention to are the training validation loss graphs.

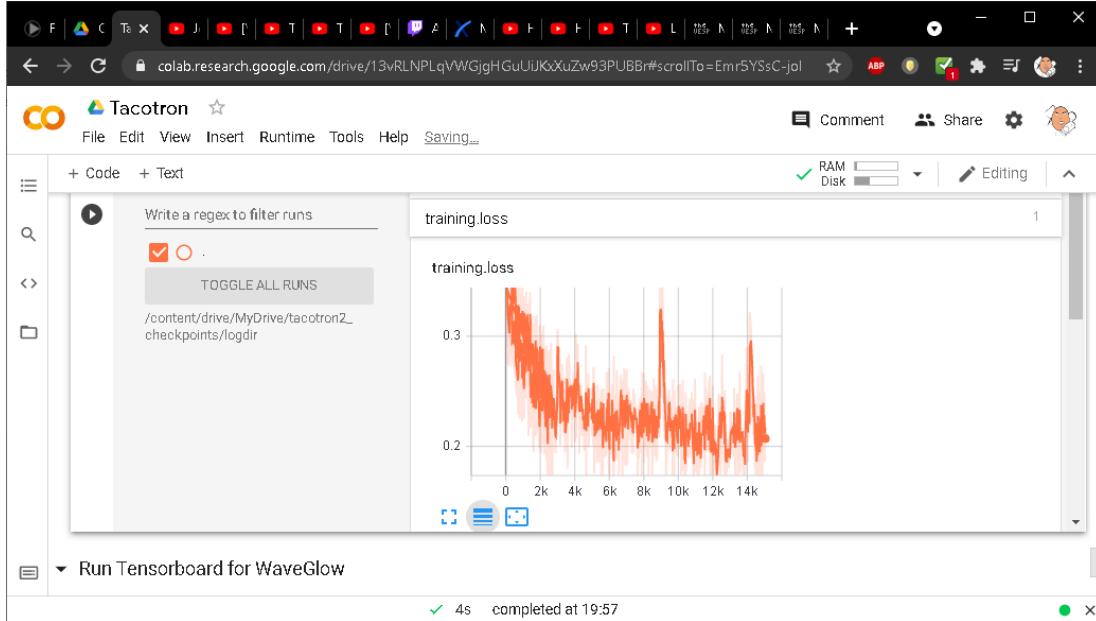


Figure 1379 - Tensorboard Scalars.

For Tacotron, once the training and validation loss curves flatten out, you should reduce the learning rate.

```
# Training list file.
!sed -i -- '$,filelists/ljs_audio_text_train_filelist.txt,/content/drive/MyDrive/tacotron2' hparams.py

# Validation list file.
!sed -i -- '$,filelists/ljs_audio_text_val_filelist.txt,/content/drive/MyDrive/tacotron2' hparams.py

# Increase epochs
!sed -i -- '$,epochs=500,epochs=10000,g' '/content/tacotron2/hparams.py'

# Save checkpoints every 200 steps.
!sed -i -- '$,iters_per_checkpoint=1000,iters_per_checkpoint=200,g' '/content/tacotron2/hparams.py'

# Reduce batch_size. If you get an 'Out Of Memory' error, reduce this further
!sed -i -- '$,batch_size=64,batch_size=35,g' '/content/tacotron2/hparams.py'

# Reduce learning rate
!sed -i -- '$,learning_rate=1e-3,learning_rate=1e-4,g' '/content/tacotron2/hparams.py'
```

Figure 1380 - Reducing learning rate.

You can do this back in the **Configure hparams.py** section by uncommenting the last line then running that code cell again.

Generally, for datasets based on Skyrim voice acting, I train for 20,000 iterations before lowering the learning rate then train for another 5,000 to 10,000 iterations. Beyond that I don't notice much improvement.

You can check the alignment in the Images section.

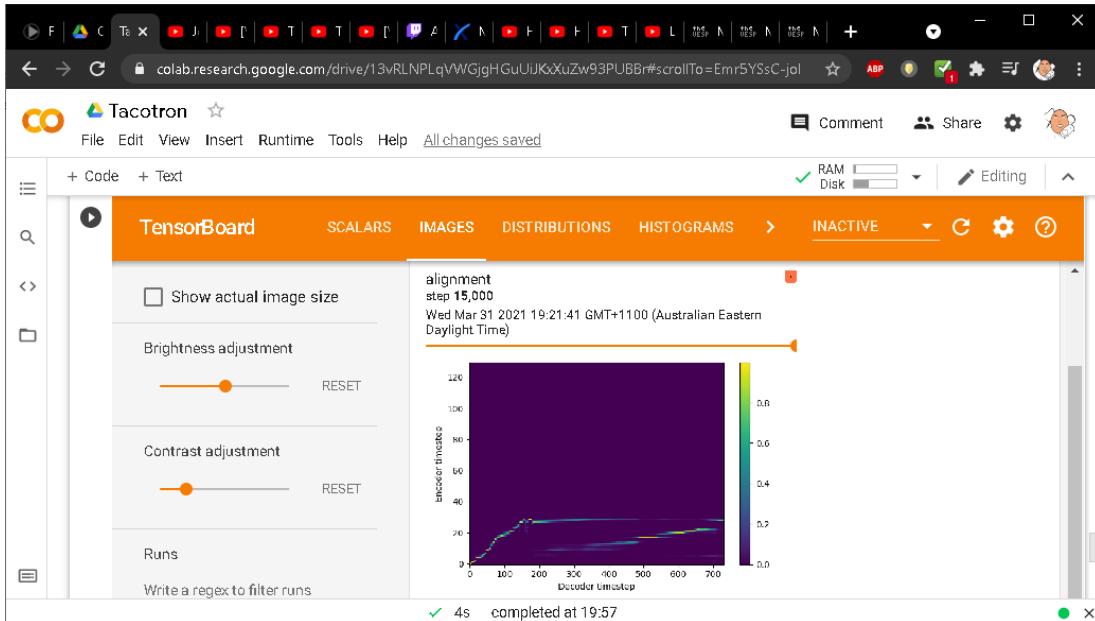


Figure 1381 - Alignment.

Alignment indicates how well a sound generated by the decoder matches a character read by the encoder.

Good alignment will look more like a horizontal line from the bottom left to the top right.

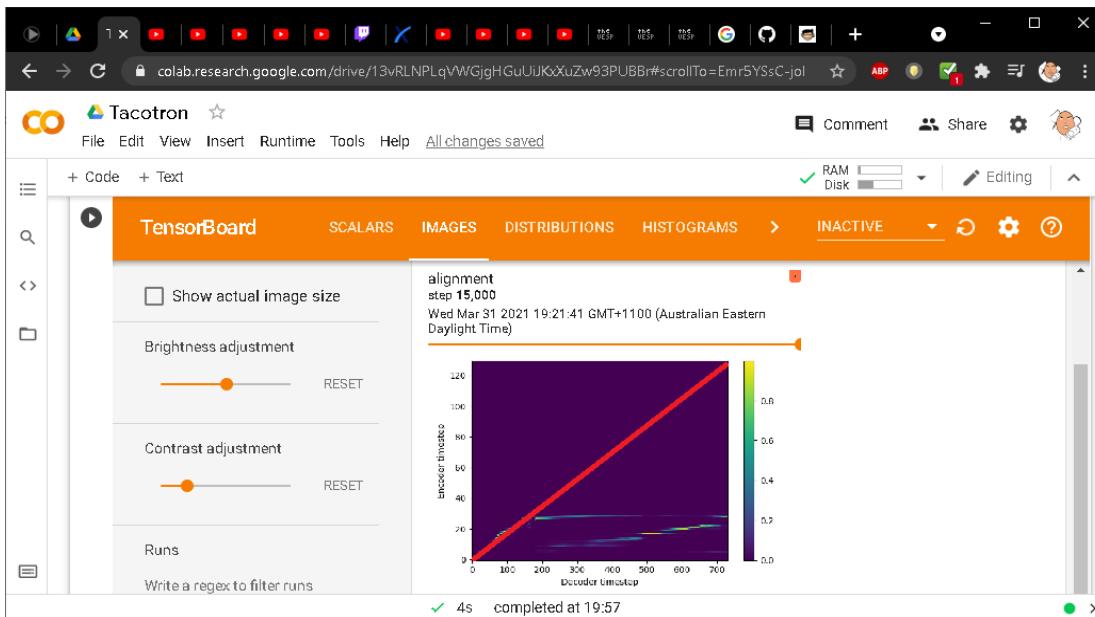


Figure 1382 - What good alignment should look like.

SYNTHESIZING AUDIO FROM THE MODELS WE'VE TRAINED

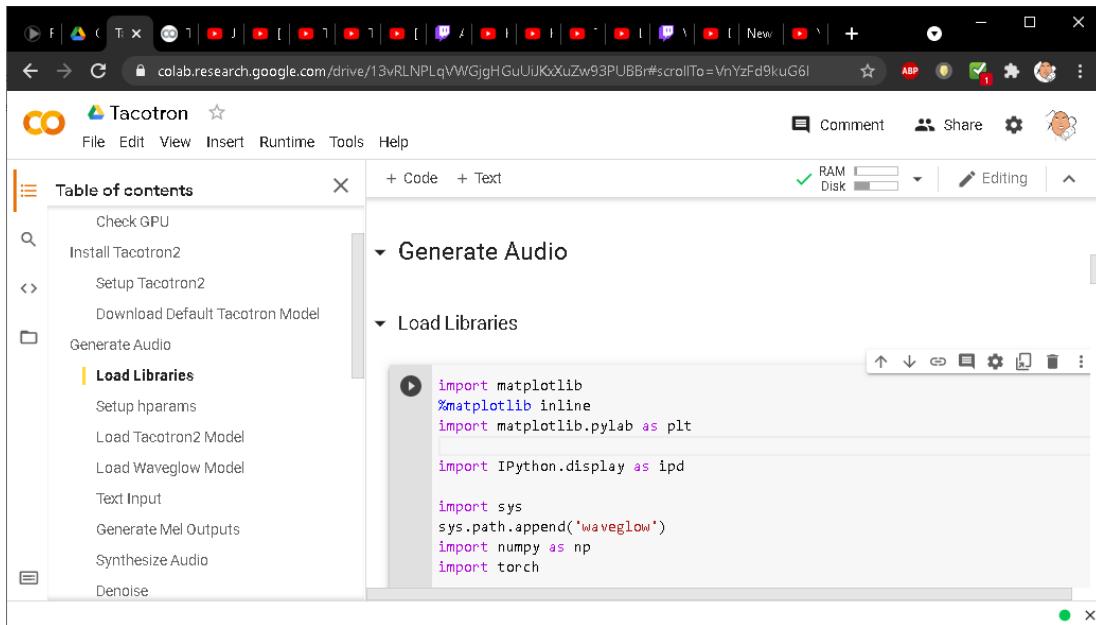
Scroll down to the section labelled **Install Tacotron 2**.

Run the code cell under **Setup Tacotron 2**. This will install and Tacotron 2 and its dependencies to your Colab session.

```
[ ] %cd /content/  
# Install Tacotron2  
!git clone https://github.com/NVIDIA/tacotron2.git  
%cd tacotron2  
# Initialize submodules  
!git submodule init; git submodule update  
# Install dependencies  
!pip install tensorflow==1.15.2  
!pip install -q unidecode tensorboardx
```

Figure 1383 - Installing Tacotron 2.

Now scroll down to **Generate Audio** and run the two code cells under **Load Libraries**.



```

import matplotlib
%matplotlib inline
import matplotlib.pyplot as plt

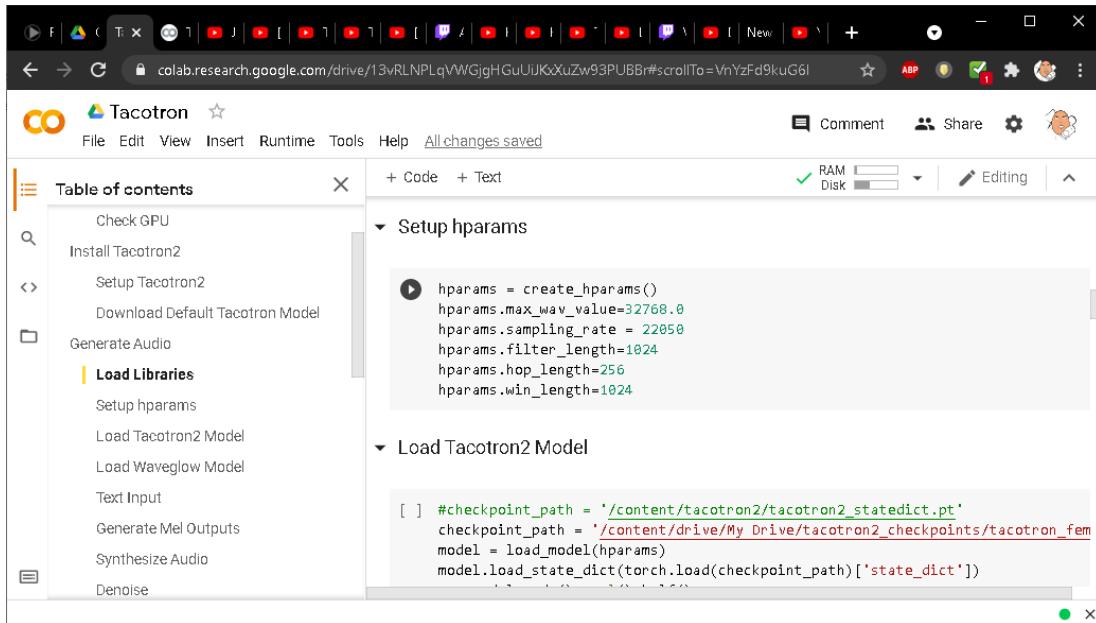
import IPython.display as ipd

import sys
sys.path.append('waveglow')
import numpy as np
import torch

```

Figure 1384 - Load libraries.

Run the code cell under **Setup hparams**.



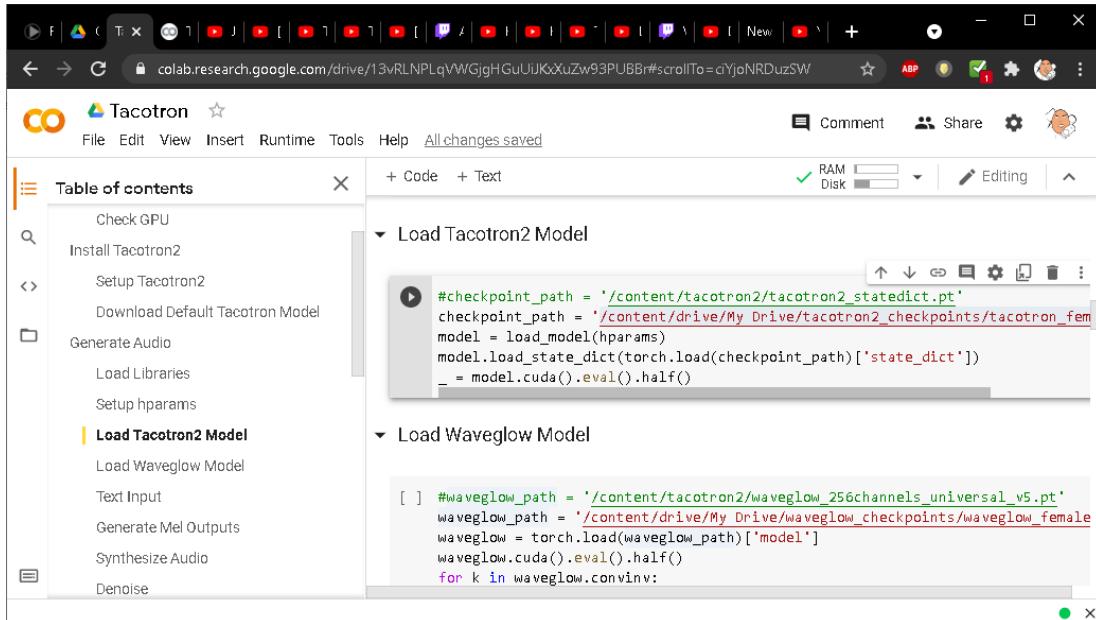
```

hparsms = create_hparams()
hparsms.max_wav_value=32768.0
hparsms.sampling_rate = 22050
hparsms.filter_length=1024
hparsms.hop_length=256
hparsms.win_length=1024

```

Figure 1385 - Setup hparams.

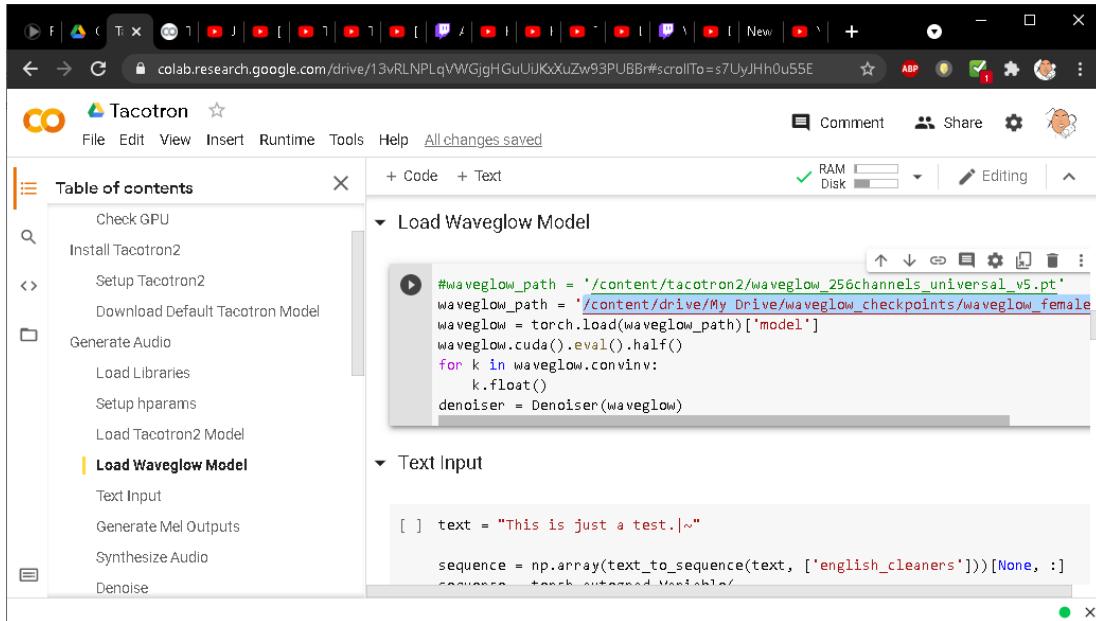
Under **Load Tacotron2 Model**, set ‘checkpoint_path’ to point to your Tacotron model before running the code cell. For my femalenord example, that’s going to be ‘/content/drive/My Drive/tacotron2_checkpoints/tacotron_femalenord’.



```
#checkpoint_path = '/content/tacotron2/tacotron2_statedict.pt'
checkpoint_path = '/content/drive/My Drive/tacotron2_checkpoints/tacotron_femalenord'
model = load_model(hparams)
model.load_state_dict(torch.load(checkpoint_path)['state_dict'])
_ = model.cuda().eval().half()
```

Figure 1386 - Load Tacotron2 Model.

Now under **Load WaveGlow Model**, set ‘waveglow_path’ to point to your WaveGlow model before running the code cell. For my femalenord example, that’s going to be ‘/content/drive/My Drive/waveglow_checkpoints/waveglow_femalenord’.



```
#waveglow_path = '/content/tacotron2/waveglow_256channels_universal_v5.pt'
waveglow_path = '/content/drive/My Drive/waveglow_checkpoints/waveglow_femalenord'
waveglow = torch.load(waveglow_path)['model']
waveglow.cuda().eval().half()
for k in waveglow.convinv:
    k.float()
denoiser = Denoiser(waveglow)
```

Figure 1387 - Load WaveGlow Model.

Under **Text Input** you specify the line of dialogue you want to generate then click run cell. Try to keep this down to one or two sentences at most. You can always edit the dialogue together in an audio editing program like Audacity later on.

I marked the end of the line with '|~'. Usually this isn't necessary but it helps tell Tacotron to stop once it reaches that character, just in case your model tends to produce gibberish at the end of dialogue.

```

text = "This is just a test.|~"
sequence = np.array(text_to_sequence(text, ['english_cleaners']))[None, :]
sequence = torch.autograd.Variable(torch.from_numpy(sequence)).cuda().long()

[14] mel_outputs, mel_outputs_postnet, _, alignments = model.inference(sequence)
plot_data((mel_outputs.float().data.cpu().numpy()[0],
           mel_outputs_postnet.float().data.cpu().numpy()[0],
           alignments.float().data.cpu().numpy()[0].T))

```

Figure 1388 - Text Input.

Run the code cell under **Generate Mel Outputs**.

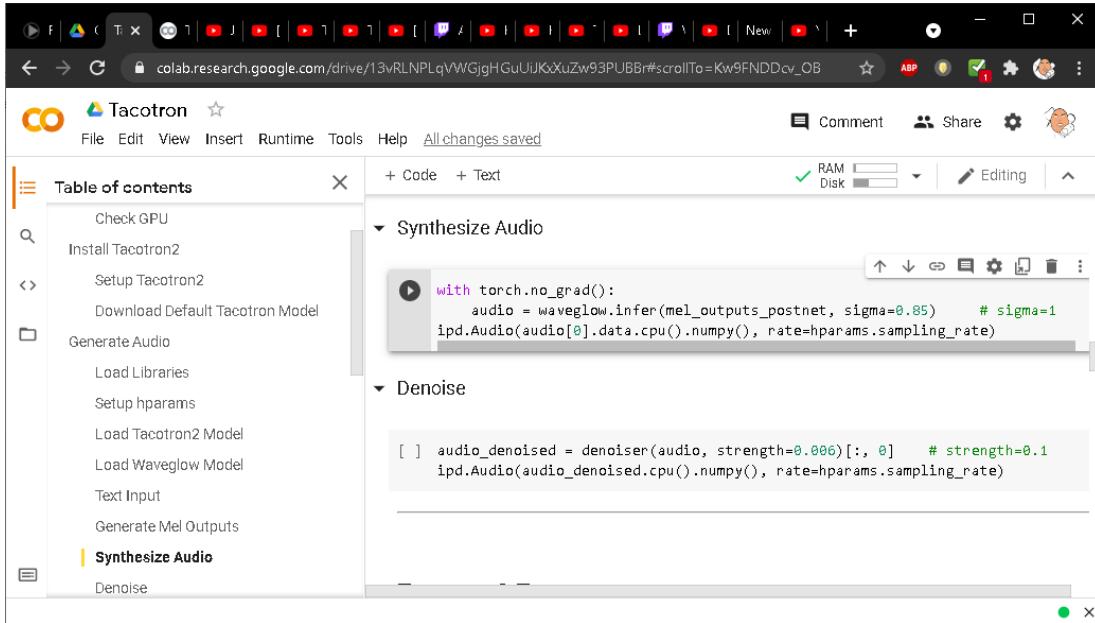
```

mel_outputs, mel_outputs_postnet, _, alignments = model.inference(sequence)
plot_data((mel_outputs.float().data.cpu().numpy()[0],
           mel_outputs_postnet.float().data.cpu().numpy()[0],
           alignments.float().data.cpu().numpy()[0].T))

```

Figure 1389 - Generate Mel Outputs.

Running the code cell under **Synthesize Audio** will generate raw unprocessed output. Press the play button to listen to the audio. You'll most likely notice some high pitched humming. The default sigma value is '1'. Lowering this value can help remove some robotic-like sounds but note that the further you lower it the more muffled the voice will be. I usually keep this set to '0.85', but you might need to lower it to '0.83' depending on how many distortions you hear in the voice.



The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar contains a "Table of contents" with various sections like "Check GPU", "Install Tacotron2", "Setup Tacotron2", "Download Default Tacotron Model", "Generate Audio", "Load Libraries", "Setup hparams", "Load Tacotron2 Model", "Load Waveglow Model", "Text Input", "Generate Mel Outputs", "Synthesize Audio", and "Denoise". The main area shows two code cells. The first cell, under "Synthesize Audio", contains:

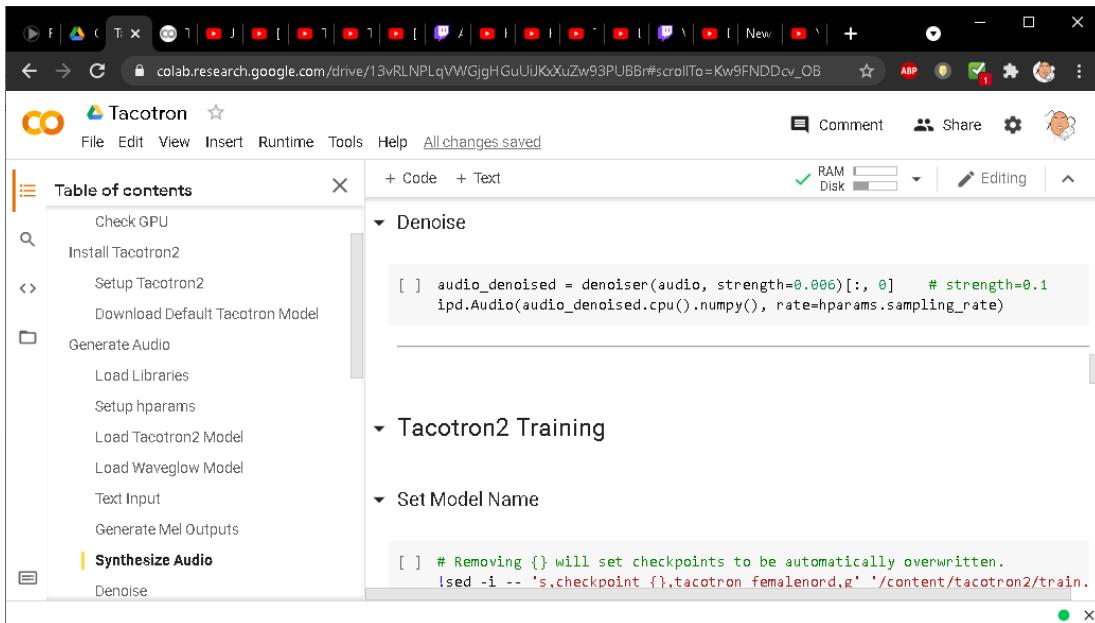
```
with torch.no_grad():
    audio = waveglow.infer(mel_outputs_postnet, sigma=0.85)      # sigma=1
ipd.Audio(audio[0].data.cpu().numpy(), rate=hparams.sampling_rate)
```

The second cell, under "Denoise", contains:

```
[ ] audio_denoised = denoiser(audio, strength=0.006)[ :, 0]    # strength=0.1
ipd.Audio(audio_denoised.cpu().numpy(), rate=hparams.sampling_rate)
```

Figure 1390 - Synthesize Audio.

We'll need to run the **Denoise** code cell to clean things up.



The screenshot shows the same Google Colab notebook. The "Table of contents" sidebar is identical. The main area shows the "Denoise" code cell from Figure 1390. The code is identical:

```
[ ] audio_denoised = denoiser(audio, strength=0.006)[ :, 0]    # strength=0.1
ipd.Audio(audio_denoised.cpu().numpy(), rate=hparams.sampling_rate)
```

Below this, under "Tacotron2 Training", there is another code cell:

```
[ ] # Removing {} will set checkpoints to be automatically overwritten.
lsed -i -- 's,checkpoint {},tacotron femalenord,g' '/content/tacotron2/train.'
```

Figure 1391 - Denoise.

The higher the strength of the denoiser the more muffled your audio will sound so try and keep this value just low enough to remove the high pitched buzzing. I usually keep the strength set to '0.006'. The default is '0.1' but I think this is a bit excessive.

You can download the audio as a .wav file by clicking on the three dots and selecting **Download**.

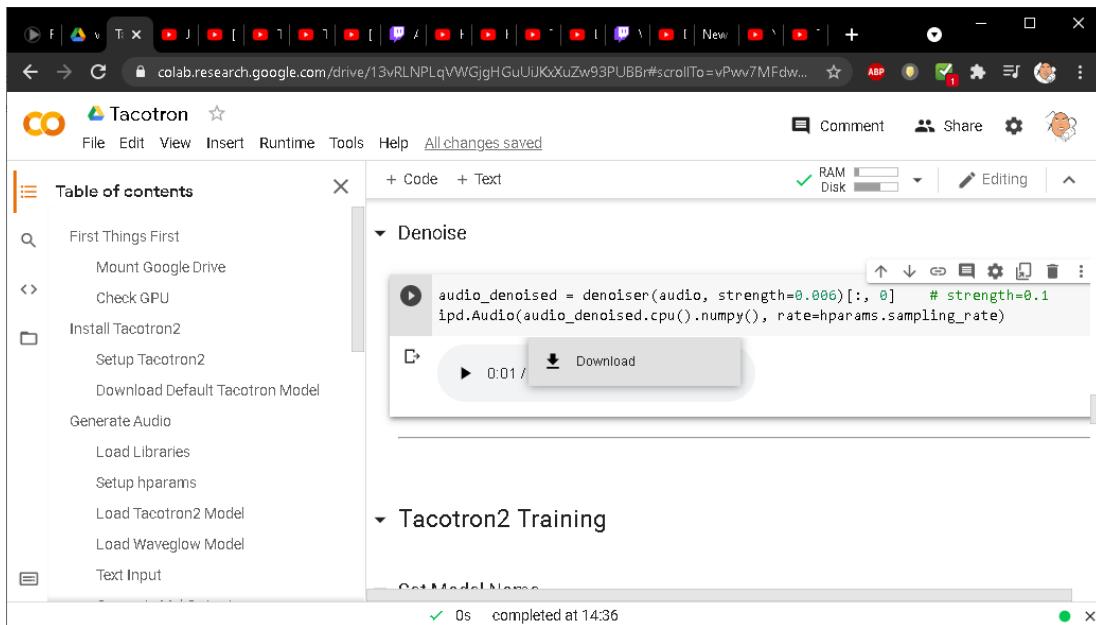


Figure 1392 - Downloading our audio.

To generate a new line of dialogue, just run the cells from **Text Input** onwards.

Now that we've generated some audio samples, it's time to try and improve audio quality in Audacity.

IMPROVING AUDIO QUALITY IN AUDACITY

In this section I'll be using [Audacity](#) to try and upsample the audio clips generated by Tacotron. You can download Audacity for free from its website.

Audio output from Tacotron will be at 22050Hz by default. While we can output audio at a higher sample rate, it likely won't sound much better.

Upsampling alone won't do much to improve the quality of our audio. We need to try and boost some of the higher frequencies to add a bit more depth into the voice.

Important: There's no *good* way to do this and there are probably better methods than what I'm about to show you. But this is just the method that I use to upsample vocal recordings.

There are AI-based audio upsampling programs out there, but they tend to introduce more distortions so I'm not going to cover that in this tutorial.

Open the .wav file in Audacity.

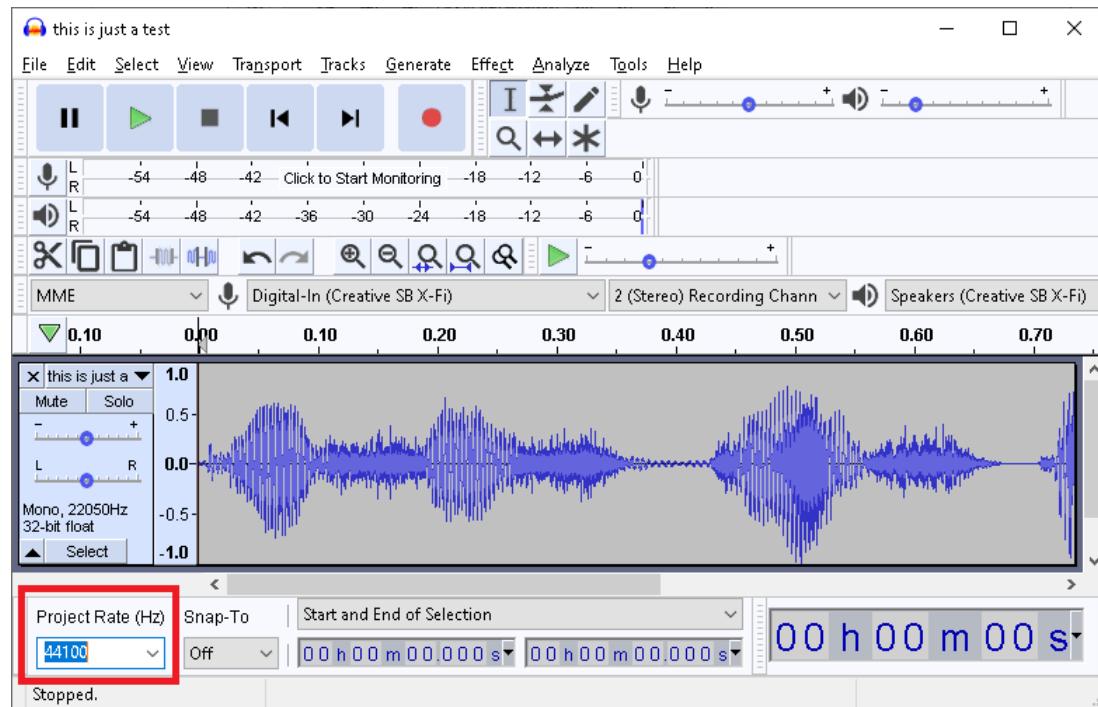


Figure 1393 - Changing the Project Rate.

Set the Project Rate (Hz) in the bottom-left corner to 44100.

Press CTRL+A to select the entire audio track.

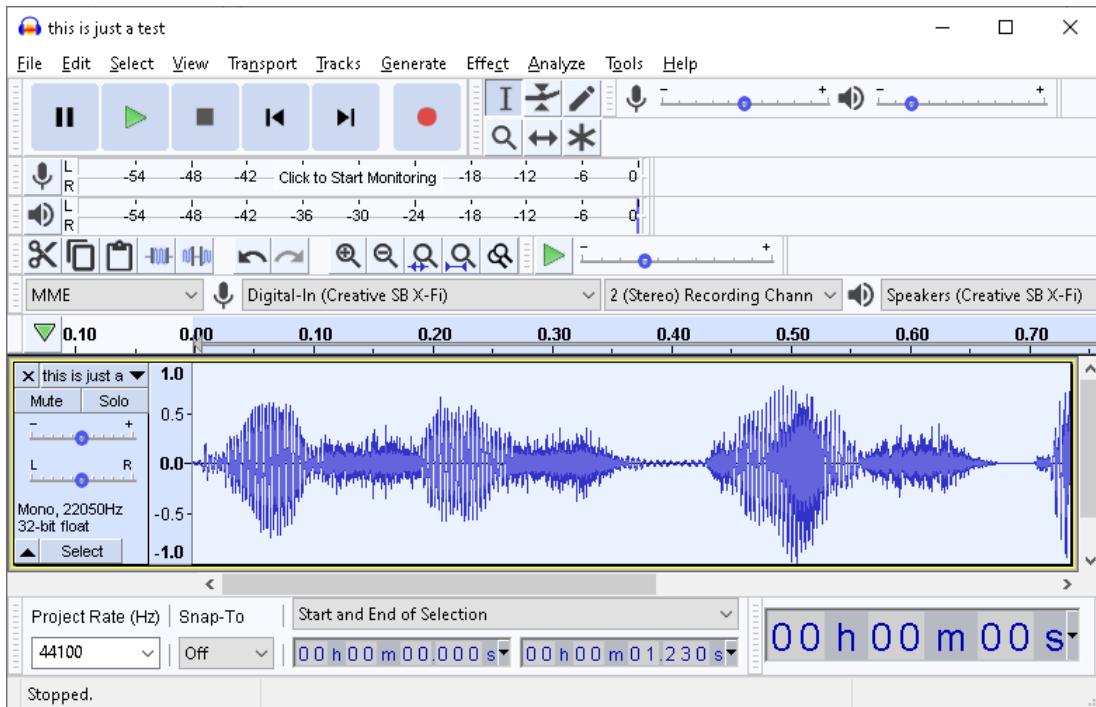


Figure 1394 - Entire audio track selected.

Now go to Tracks > Resample...

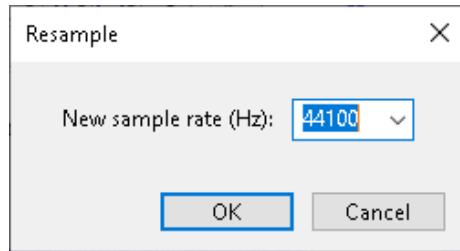


Figure 1395 - Resampling to 44100 Hz.

Make sure the 'New sample rate (Hz)' is set to 44100 then click OK.

Go to Edit > Duplicate to copy the selected audio track.

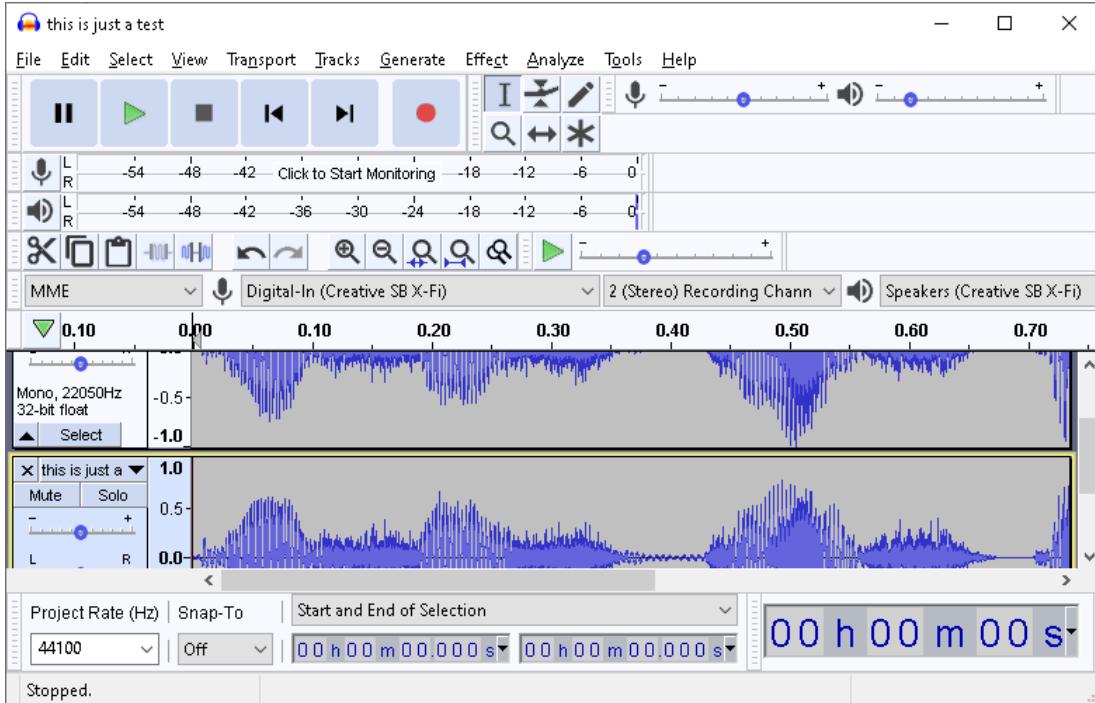


Figure 1396 - Audio track duplicated.

Select the entirety of the duplicate track by double-clicking anywhere within it.

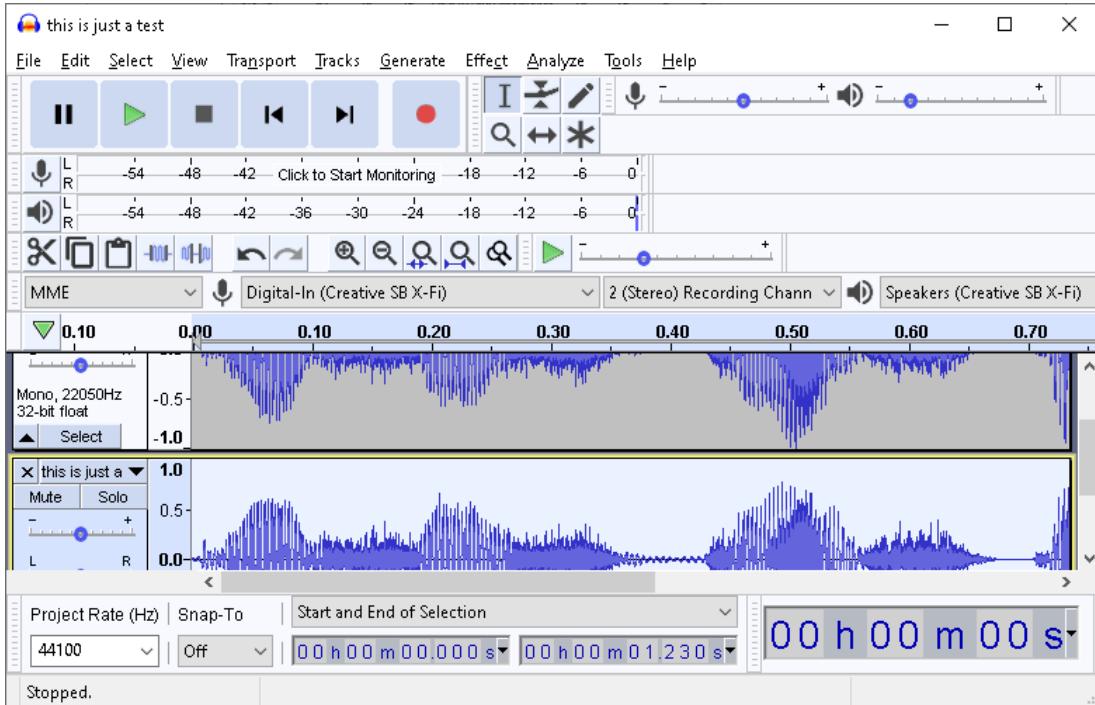


Figure 1397 - Duplicate track selected.

Go to Effect > High Pass Filter...

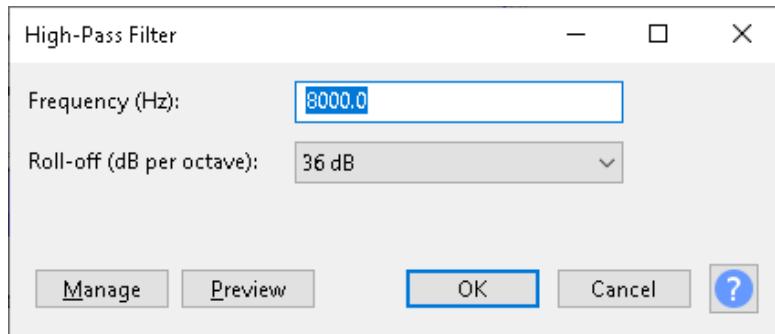


Figure 1398 - High Pass Filter.

Set 'Frequency (Hz)' to 8000.0 and 'Roll-off (dB per octave)' to 36 dB then click OK. This will isolate just the higher frequencies.

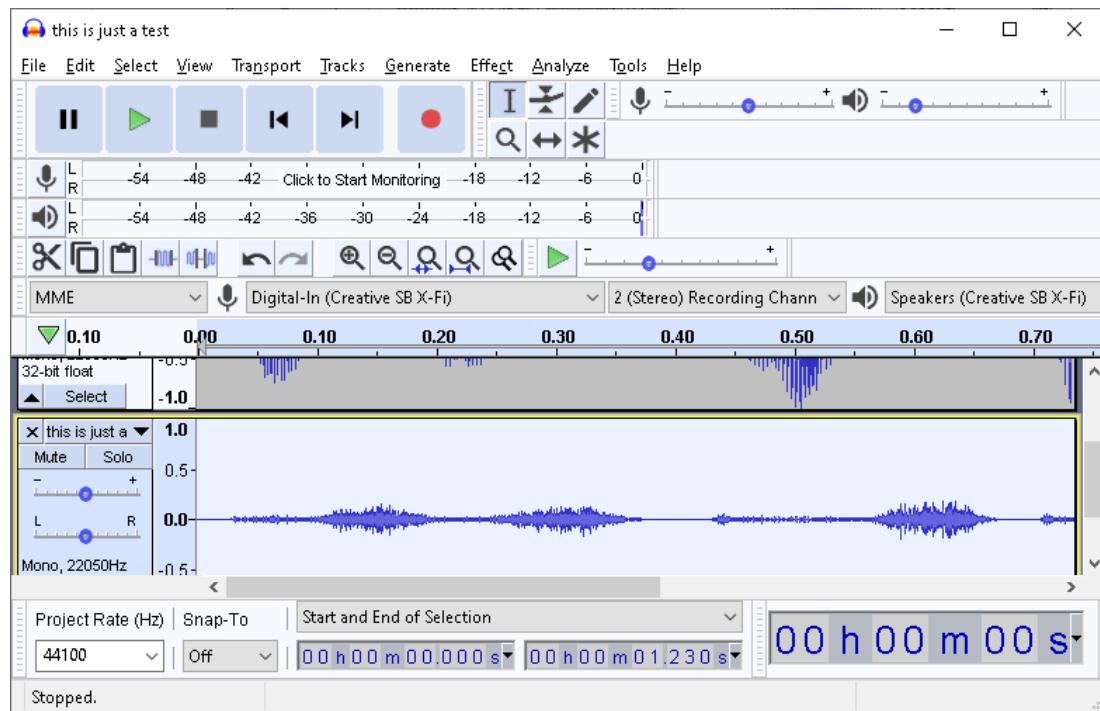


Figure 1399 - High Pass Filter applied to duplicate track.

We need to raise the pitch of these higher frequencies a bit. With the duplicate track still selected, go to Effect > Change Pitch...

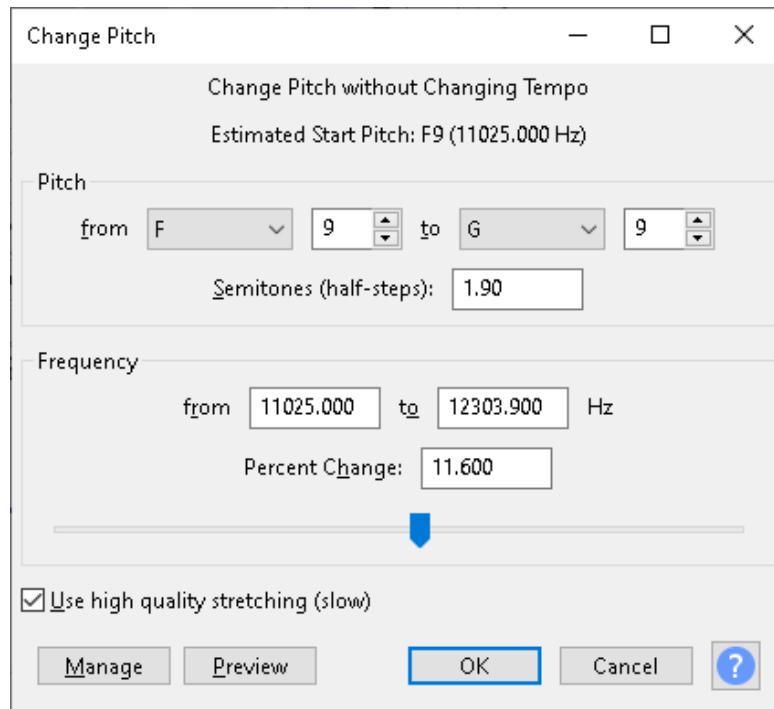


Figure 1400 - Change Pitch.

Set 'Semitones (half-steps)' to 1.90. Ensure 'Use high quality stretching (slow)' is ticked then click OK.

The last thing we need to do is play around with the gain of the duplicate track. For male voices I usually set this to around +3 dB depending on the voice, just to add a little bit more depth. For female voices I set this to -3 dB to lessen high pitched hissing.

You'll need to play this by ear and find out what sounds best yourself.

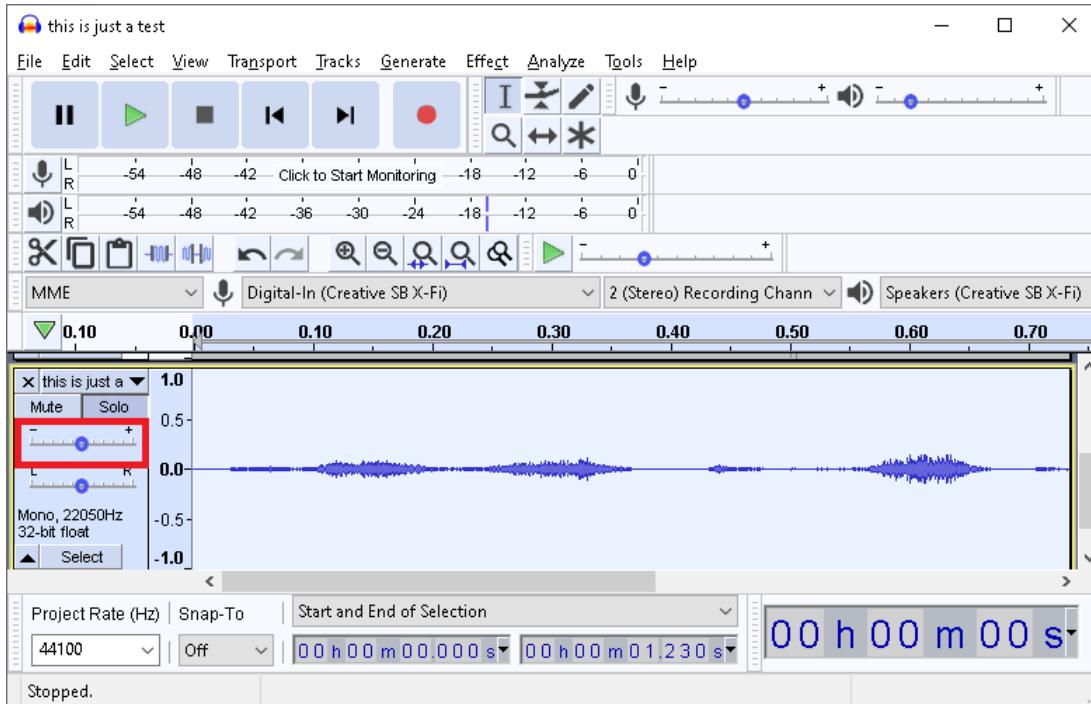


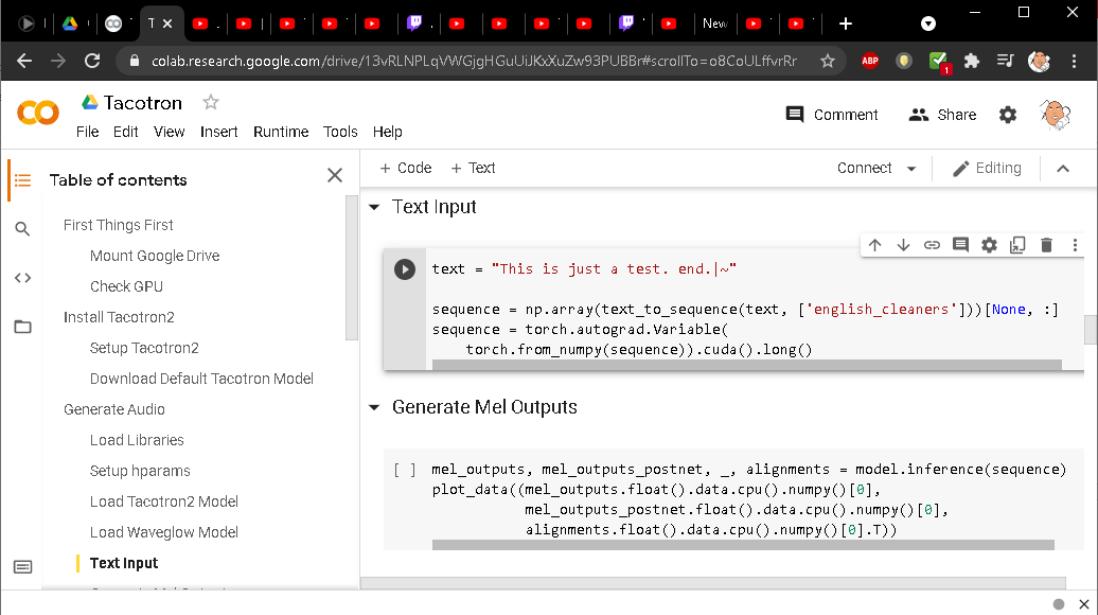
Figure 1401 - Gain controls.

To save a new .wav file, go to File > Export > Export as WAV.

Select the folder you want to save the .wav file to then click Save.

A COUPLE TIPS AND TRICKS

TIP 1: Sometimes part of the last word will get cut off, so what I like to do is add another word after it, like 'end', to help mitigate that.



The screenshot shows a Google Colab notebook titled "Tacotron". On the left, there's a "Table of contents" sidebar with sections like "First Things First", "Install Tacotron2", and "Generate Audio". The main area has a "Text Input" section expanded, showing Python code. The code defines a variable `text` with the value "This is just a test. end.|~". It then creates a NumPy array `sequence` from this text, converts it to a PyTorch Variable, and moves it to the GPU. Below this, another section "Generate Mel Outputs" is partially visible, showing code for generating mel spectrograms and alignments.

```

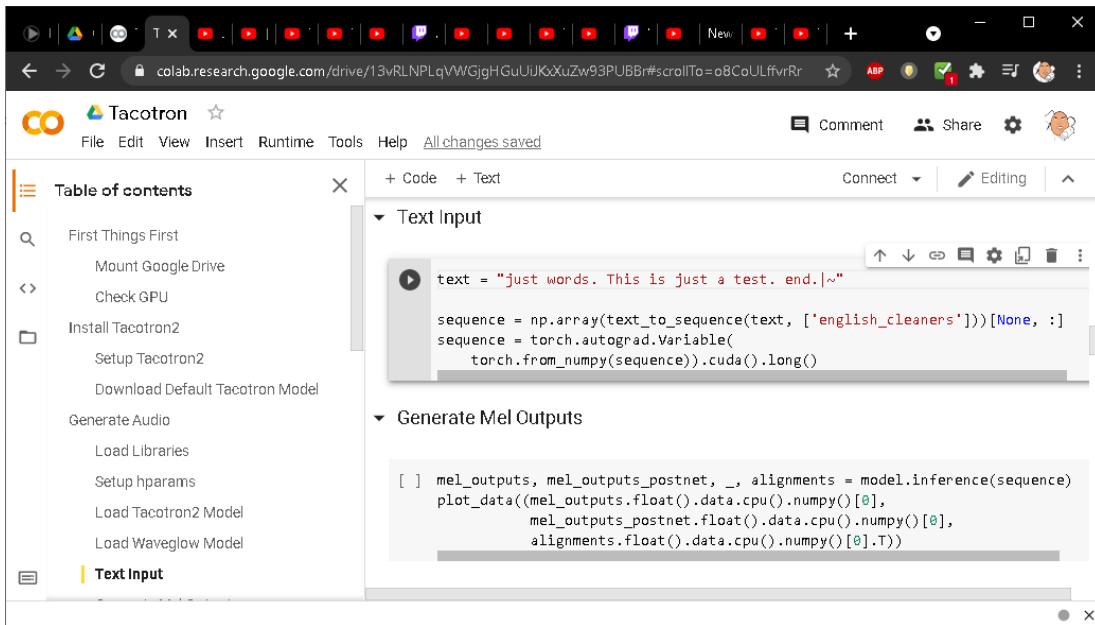
text = "This is just a test. end.|~"
sequence = np.array(text_to_sequence(text, ['english_cleaners']))[None, :]
sequence = torch.autograd.Variable(
    torch.from_numpy(sequence)).cuda().long()

```

Figure 1402 - Ending word.

TIP 2: The first few words seem to have a large impact on the delivery of the rest of the dialogue.

'this is just a test' may sound a lot different to 'elf scum! this is just a test'. If you copy part of an existing line of dialogue from your dataset, like 'never should've come here!' and prefix it before the sentence you actually want to generate, some of the emotional intensity from the original line of dialogue will transfer across to the new line of dialogue.



The screenshot shows a Google Colab notebook titled "Tacotron". The left sidebar contains a "Table of contents" with various sections like "First Things First", "Install Tacotron2", and "Generate Audio". The main area has two code cells. The top cell is titled "Text Input" and contains the following Python code:

```
text = "just words. This is just a test. end.|~"
sequence = np.array(text_to_sequence(text, ['english_cleaners']))[None, :]
sequence = torch.autograd.Variable(
    torch.from_numpy(sequence)).cuda().long()
```

The bottom cell is titled "Generate Mel Outputs" and contains:

```
[ ] mel_outputs, mel_outputs_postnet, _, alignments = model.inference(sequence)
plot_data((mel_outputs.float().data.cpu().numpy()[0]),
          mel_outputs_postnet.float().data.cpu().numpy()[0],
          alignments.float().data.cpu().numpy()[0].T)
```

Figure 1403 - Starting words.

If your dialogue is being screamed or shouted and you don't want it to be, try prefixing something benign like 'hello there' and that should lower the emotional intensity of the rest of the dialogue.

Capitalization doesn't really make a difference to the output.

TIP 3: Every time you generate dialogue it'll sound different. If you don't like the way a specific sentence was conveyed, just run the code cells from **Generate Mel Outputs** onwards and check again.

If I hear some robotic-like noises or distortions, I just generate the dialogue over and over until I get a clean read.

TIP 4: Not all words will be pronounced properly. The malenordcommander model I trained has issues with words like 'dialogue'. In those cases I'd have to type in 'die a log' instead. For making malenordcommander pronounce 'python' properly (and not as 'pee-thon') I'd have to type it in as 'piiethon'.

It'll require a bit of trial and error to figure out what works in those cases.