Queensland Prosidy - Exploring the sounds of Queensland voices through machine learning

SLQ Wiki Fabrication Lab 2025/07/05 19:11

This workshop is a bit of a provocation, intended to lead into deeper engagement with State Library's original oral history collection with the potential for a community developed AI Queensland Voice(s). It cover the existing state of the art, and how even extremely large models

Queensland Prosidy - Exploring the sounds of Queensland voices through machine learning

In this workshop we'll look at how the unique accents, rhythms and tones of Queensland voices from State Library's collection fit into the burgeoning world of Text To Speech (TTS). We'll search the collection for some original audio sources, explore the resources available at The Edge for audio restoration and finally learn how to use free and open source machine learning software to create a realistic voice. We'll cover briefly the legal and ethical background to 'voice fakes' - and discover how hard it really is to make a machine speak Strine.

Voices in the Collection

How to find them? Digitised oral histories are a good bet, meaning a transcript should be available.

- “oral history digital”
- refine by “original materials”
- sort by date - “oldest to newest”

Lets go with164 oral history interviews regarding the history of the Cape York Peninsula by interviewer Duncan Jackson

Finding Digital Audio and transcripts

Lets have a listen to Duncan Jackson's interview with Kathleen Jackson.

Click on online access to access the digitool viewer, then open:

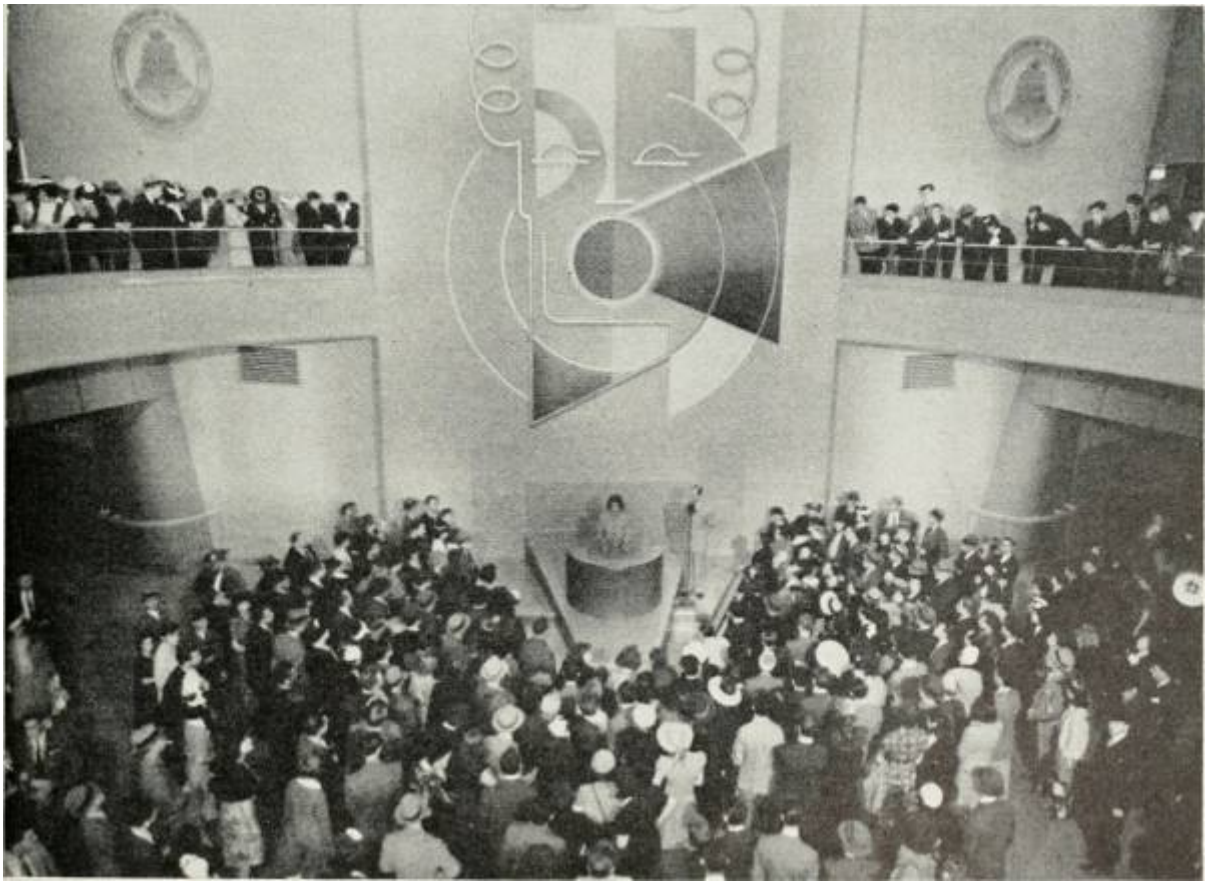
- mp3
- PDF transcript

Copyright and ethical considerations

We can always find the conditions of access and use on onesearce and the digitool viewer. In this case we have unrestricted access, and the material is in copyright.

Speech Synthesis

Like many of the 20th century's technological inovations, the frst modern speech synthesiser can be traced back to the invention of the vocoder at Bell Labs. Derived from this, the Voder was demonstrated at the 1939 World Fair.



1)

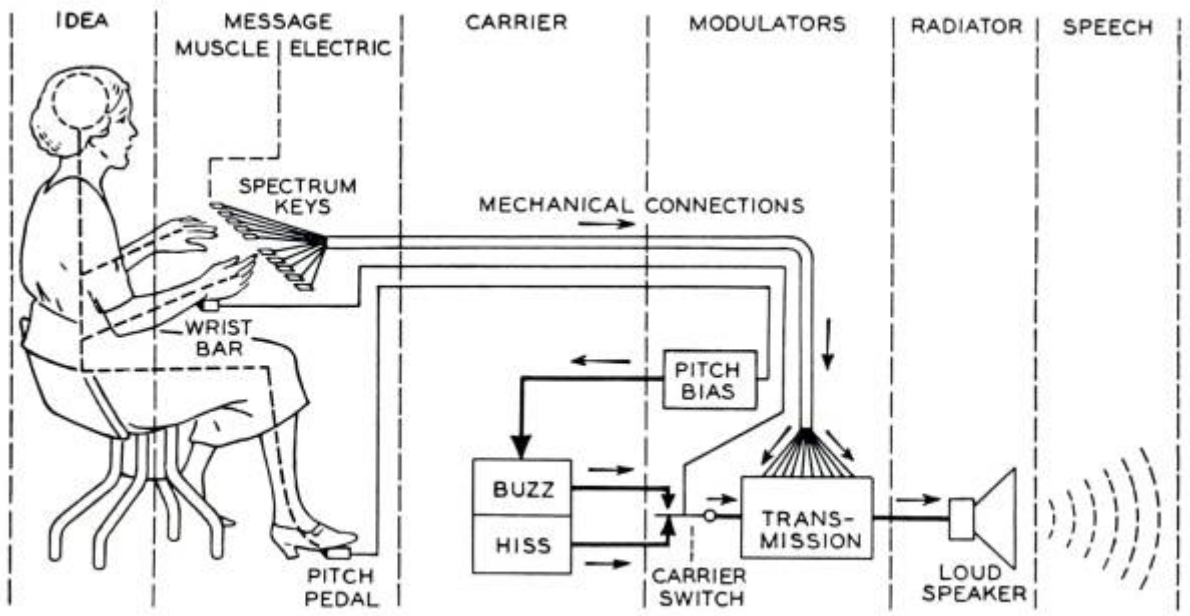


Fig. 8—Schematic circuit of the voder.

Historical Audio Examples

Here is a playlist of various historical TTS methods.

https://soundcloud.com/user-552764043

Modern State of the Art TTS

Now - it time to have some fun with TTS - check out the man holding the frog below...

https://vo.codes/#speak

And have a listen to some interesting examples from pop/meme culture.

https://fifteen.ai/examples

https://www.youtube.com/watch?v=drirw-XvzzQ

Wavenet

Modern deep learning based synthesis started with the release of Wavenet in 2016 by Google's Deepmind.

WaveNet changes this paradigm by directly modelling the raw waveform of the audio signal, one sample at a time. As well as yielding more natural-sounding speech, using raw waveforms means that WaveNet can model any kind of audio, including music.[2)]

Tacotron and Tacotron2

Wavenet was followed by Tacoctron (also from Google) in 2017.

https://google.github.io/tacotron/publications/tacotron/index.html

Then Tacotron2

https://ai.googleblog.com/2017/12/tacotron-2-generating-human-like-speech.html

The next wave - Diffusion

In April 2022 open.ai dropped DALL-E 2, which uses diffusion models.

“Diffusion Models are generative models, meaning that they are used to generate data similar to the data on which they are trained. Fundamentally, Diffusion Models work by destroying training data through the successive addition of Gaussian noise, and then learning to recover the data by reversing this noising process. After training, we can use the Diffusion Model to generate data by simply passing randomly sampled noise through the learned denoising process.”

These models can be applied to TTS, with tortoise-tts producting someexcellent examples of generated speech.

Getting Started

Google Colab

Google's Colaboratory[3)], or “Colab” for short, allows you to write and execute Python in your browser, with

- Zero configuration required
- Free access to GPUs
- Easy sharing

Python

Python is an open source programming language that was made to be easy-to-read and powerful[4]). Python is:

- a high-level language, (Meaning programmer can focus on what to do instead of how to do it.)
- an interpreted language (Interpreted languages do not need to be compiled to run.)
- is often described as a “batteries included” language due to its comprehensive standard library.

A program called an interpreter runs Python code on almost any kind of computer. In our case python will be interpreted by google colab, which is based on Jupyter notebooks.

Jupyter Notebooks

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text[5]. Usually Jupyter notebooks require set-up for a specific purpose, but Colab takes care of all this for us.

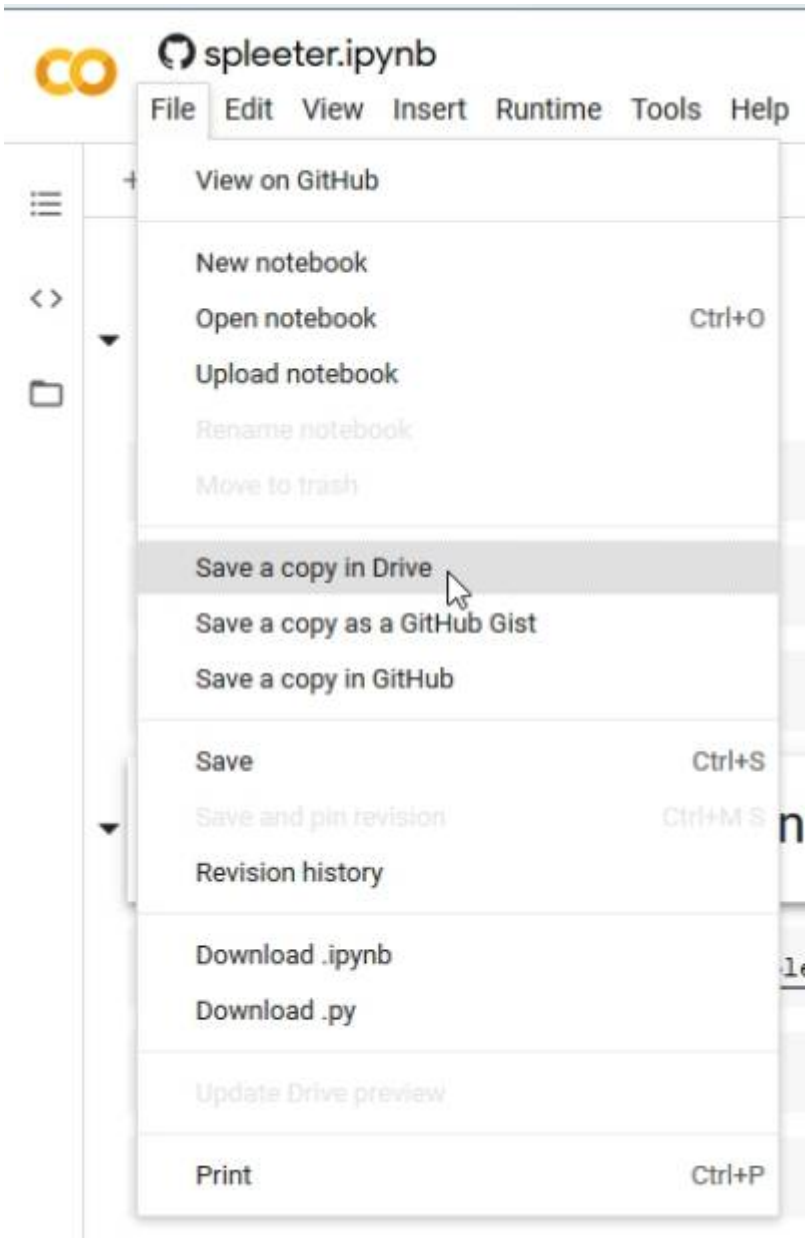
Getting Started with Colab

The only requirment for using Colab is (unsurprisingly) a Google account. Once you have a google account, lets jump into our first ML example - Spleeter - that we mentioned earlier. Go to the Colab here:

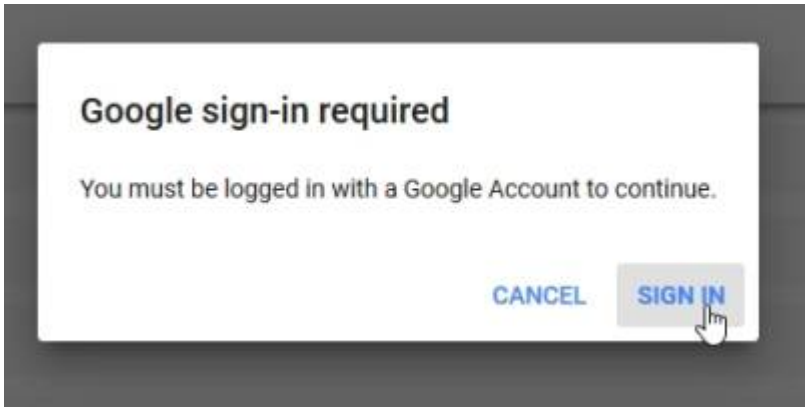
https://colab.research.google.com/github/deezer/spleeter/blob/master/spleeter.ipynb

Making a Colab Copy

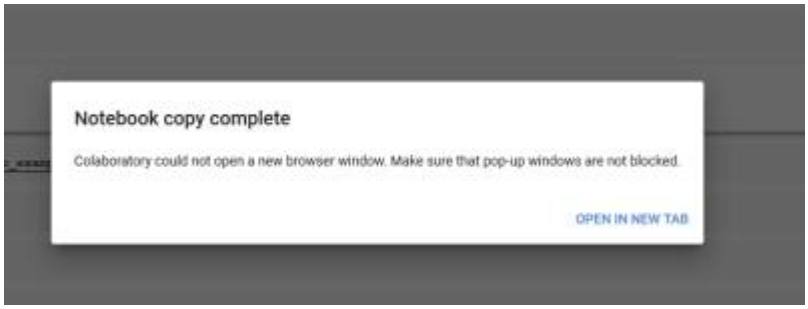
The first step is make a copy of the notebook to our Google drive - this means we can save any changes we like.



This will trigger a google sign-in

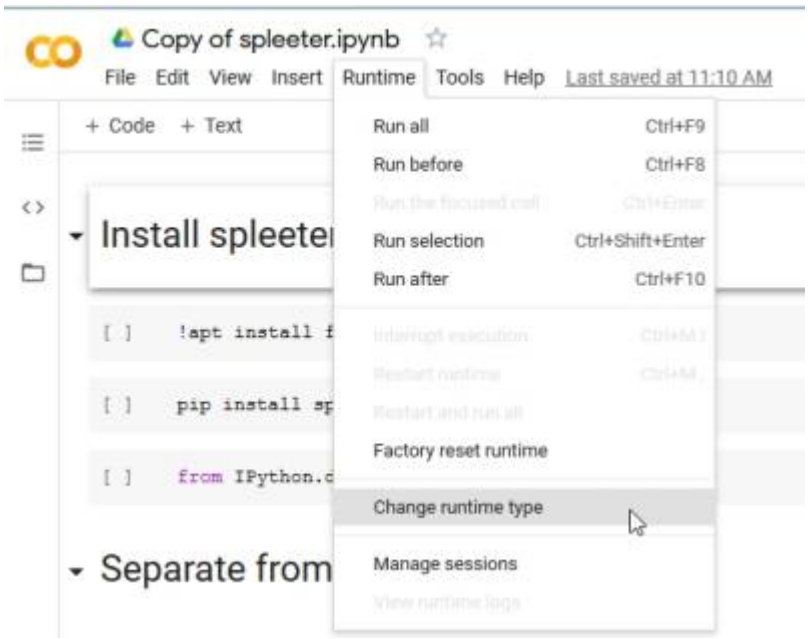


and the your copy will open in a new tab.

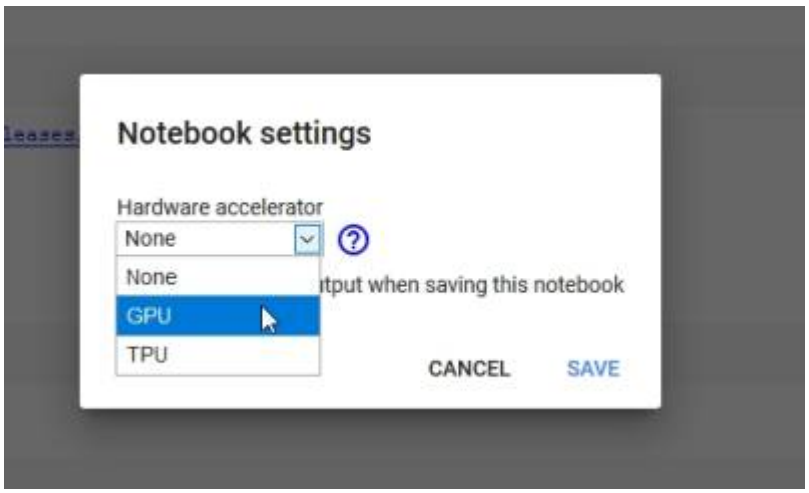


Select a Runtime

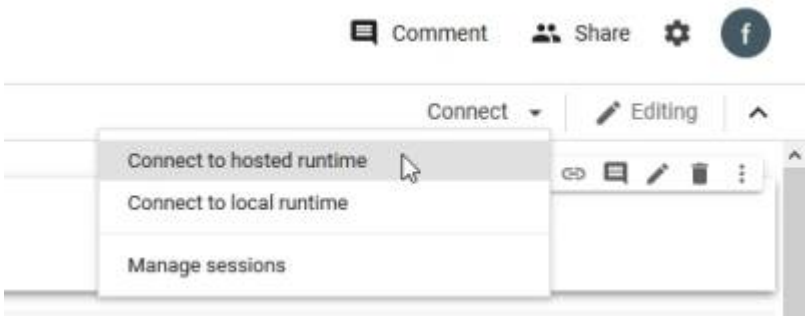
Next we change our runtime (the kind or processor we use)



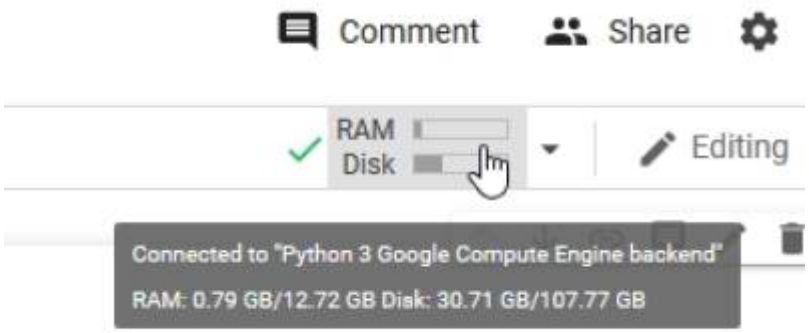
to a GPU to take advantage of Googles free GPU offer.



Now lets connect to our hosted runtime



and check the specs...



Step Through the Notebook

Now its time to actually use the notebook! Before we start, lets go over how the notebooks work:

- The notebook is divided into sections, with each section made up of cells.
- These cells have code pre-entered into them,

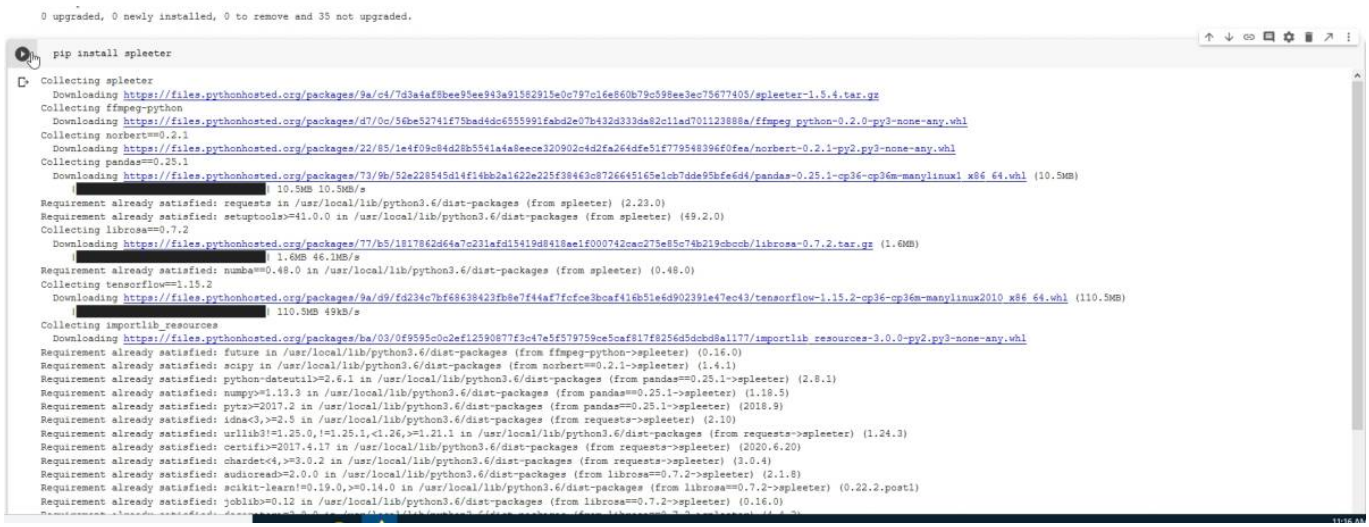
- A play button on the runs (executes) the code in the cell.
- The output of the cell is printed (or displayed) directly below each cell.
- The output could be text, pictures, audio or video.

Cells usually contain python code, but can also be coded in bash - the UNIX command line shell. Cells containing bash commands start with an exclamation mark `!`

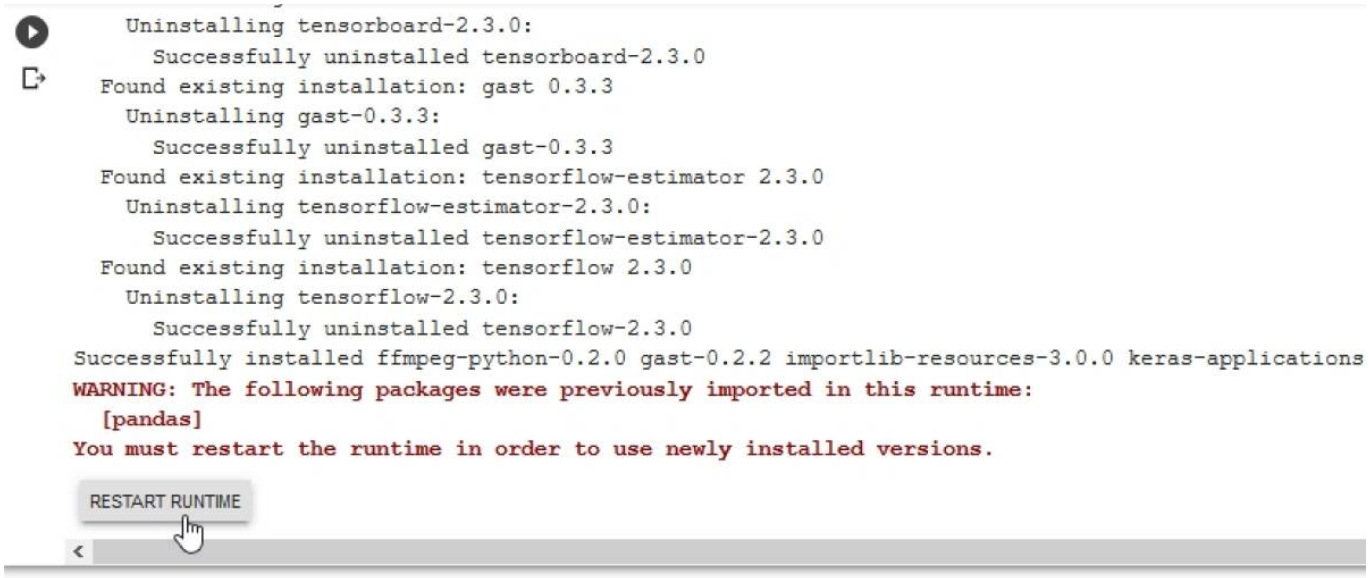
Our first section is called “Install Spleeter” and contains the bash command `apt install ffmeg` . This installs ffmeg in our runtime, which is used to process audio. Press the go button..

ffmpeg will be downloaded and installed to our runtime.

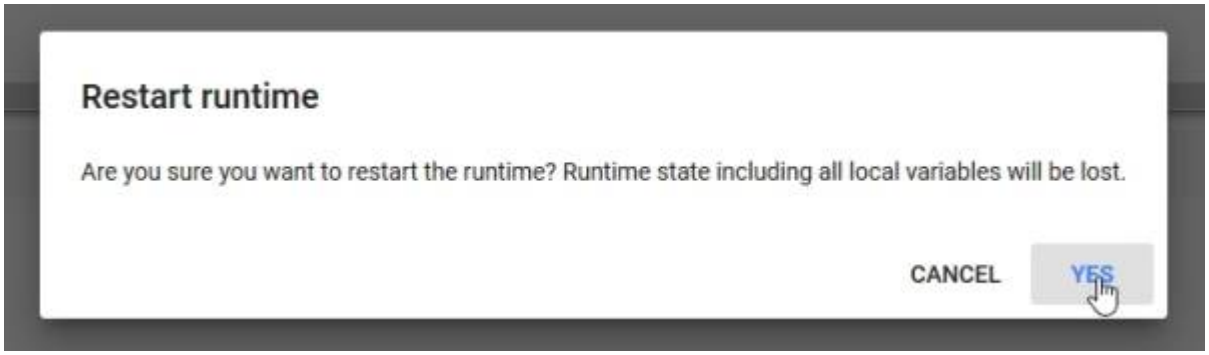
Next we will run a python command `pip`to use the python package manager to install the spleeter python package.



This will take a while - and at the end we will get a message saying we need to restart our runtime due to some compatibilty issues [6)]



Go ahead and restart



Next is another bash command

```
wget
```

we use to (web)get our example audio file.

And the next cell uses the python `Audio` command to give us a nice little audio player so we can hear our example.

Now its finally time to use the spleeter tool with the `separate` command [7] as `!spleeter separate` , and lets pass the `-h` flag [8] to show us the built in help for the command.

```
!spleeter separate -h
```

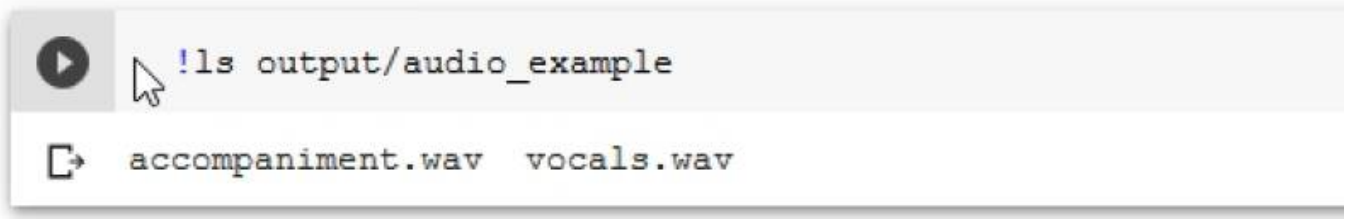
```
usage: spleeter separate [-h] [-a AUDIO_ADAPTER] [-p CONFIGURATION]
                         [--verbose] -i INPUTS [INPUTS ...] [-o OUTPUT_PATH]
                         [-f FILENAME_FORMAT] [-d DURATION] [-s OFFSET]
                         [-c {wav,mp3,ogg,m4a,wma,flac}] [-b BITRATE] [-m]
                         [-B {tensorflow,librosa,auto}]

optional arguments:
  -h, --help            show this help message and exit
  -a AUDIO_ADAPTER, --adapter AUDIO_ADAPTER
                        Name of the audio adapter to use for audio I/O
  -p CONFIGURATION, --params_filename CONFIGURATION
                        JSON filename that contains params
  --verbose             Shows verbose logs
  -i INPUTS [INPUTS ...], --inputs INPUTS [INPUTS ...]
                        List of input audio filenames
  -o OUTPUT_PATH, --output_path OUTPUT_PATH
                        Path of the output directory to write audio files in
  -f FILENAME_FORMAT, --filename_format FILENAME_FORMAT
                        Template string that will be formatted to
                        generatedoutput filename. Such template should be
                        Python formattablestring, and could use {filename},
                        {instrument}, and {codec}variables.
  -d DURATION, --duration DURATION
                        Set a maximum duration for processing audio (only
                        separate offset + duration first seconds of the input
                        file)
  -s OFFSET, --offset OFFSET
                        Set the starting offset to separate audio from.
  -c {wav,mp3,ogg,m4a,wma,flac}, --codec {wav,mp3,ogg,m4a,wma,flac}
                        Audio codec to be used for the separated output
  -b BITRATE, --birate BITRATE
                        Audio bitrate to be used for the separated output
  -m, --mwf             Whether to use multichannel Wiener filtering for
                        separation
  -B {tensorflow,librosa,auto}, --stft-backend {tensorflow,librosa,auto}
                        Who should be in charge of computing the stfts.
                        Librosa is faster than tensorflow on CPU and uses less
                        memory. "auto" will use tensorflow when GPU
                        acceleration is available and librosa when not.
```

Now that we know what we are doing - we run the tool for real, and will use the `-i` flag to define the input as our downloaded example, and the `-o` flag to define our output destination as the directory (folder) `output`. By default spleeter will download and use the2stems model.

```
!spleeter separate -i audio_example.mp3 -o output/
```

```
INFO:spleeter:Downloading model archive https://github.com/deezer/spleeter/releases/download/v1.4.0/2stems.tar.gz
INFO:spleeter:Validating archive checksum
INFO:spleeter:Extracting downloaded 2stems archive
INFO:spleeter:2stems model file(s) extracted
INFO:spleeter:File output/audio_example/accompaniment.wav written succesfully
INFO:spleeter:File output/audio_example/vocals.wav written succesfully
```

Another bash command `ls` (list) shows us the contents of our output directory



And finally onother couple of `audio` commands to hear our result!

Things to try

Check out the usage instructions for the separate tool on the Github site and try your own 4stem and 5tem separations. Use your own audio files to test the separation.

Speech to Text with Mozilla Deepspeech

Our next challenge will be to adapt the latest version of Mozilla's Deepspeech for use in Google Colab.

We will be using the documentation here:

https://deepspeech.readthedocs.io/en/v0.8.0/USING.html#getting-the-pre-trained-model

To adapt this colab notebook to run the latest version of Mozilla Deepspeech:

https://colab.research.google.com/github/tugstugi/dl-colab-notebooks/blob/master/notebooks/MozillaDeepSpeech.ipynb#scrollTo=4OAYywPHApuz

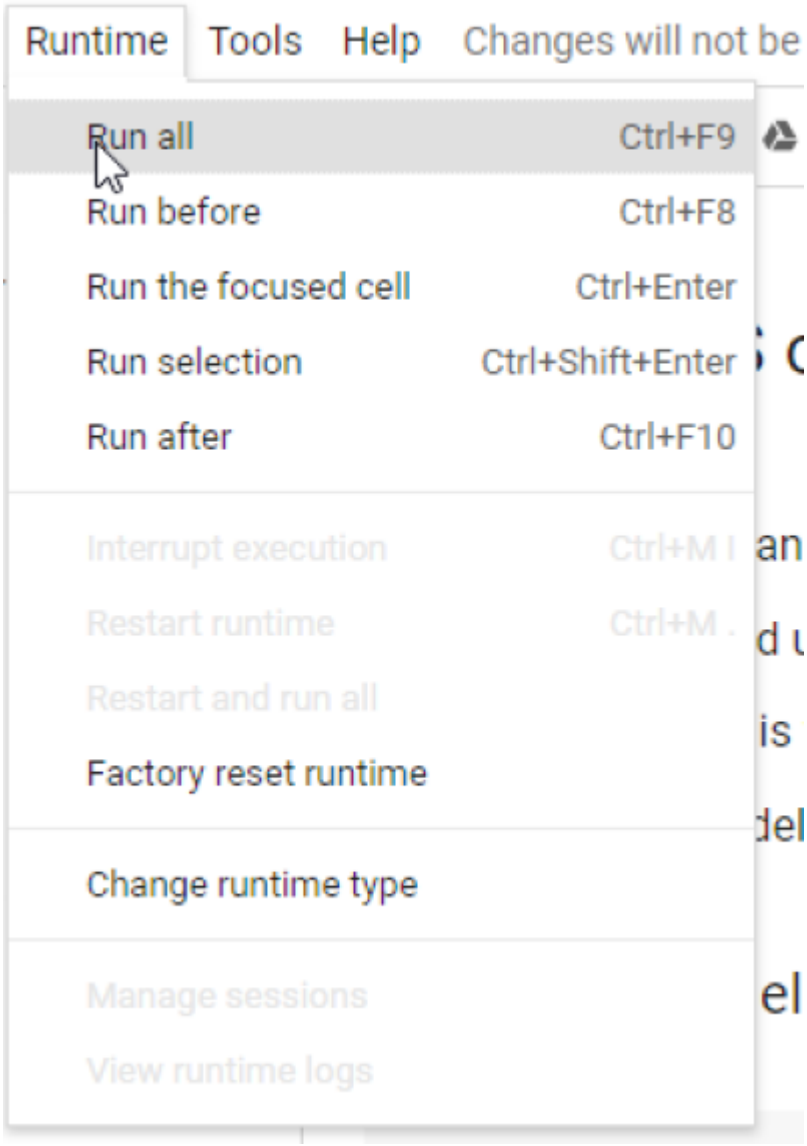
Text to Speech with Mozilla TTS

Our final example is TTS with Mozilla TTS:

https://colab.research.google.com/drive/1u_16ZzHjKYFn1HNVuA4Qf_i2MMFB9olY?usp=sharing#scrollTo=6LWsNd3_M3MP

You can dive straight into this and use it to generate speech. This example usesTacotron2 and MultiBand-Melgan models and LJSpeech dataset.

Run All Cells



Generate Speech

Going Further

ML is such a big and fast moving area of research there are countless other ways to explore and learn, here are a few two-minute videos to pique your interest:

- Video restoration
- OpenAI Plays Hide and Seek

Make sure you check out the resources in Lynda, which you will have free access to as a State Library of Queensland member

Links

https://machinelearningforkids.co.uk/#!/links#top

https://experiments.withgoogle.com/collection/ai

https://openai.com/blog/

1)
By Internet Archive Book Images - https://www.flickr.com/photos/internetarchivebookimages/14776509983/Source book page: https://archive.org/stream/belltelephonemag19amerrich/belltelephonemag19amerrich#page/n78/mode/1upReference[Fig.4] The Voder Fascinates the Crowds from: Williams, Thomas W. (January 1940) I. At the New York World's Fair. "Our Exhibits at Two Fairs". Bell Telephone Quarterly XIX (1): 65."The Voder Fascinates the Crowds - The manipulative skill of the operator s fingers makes the Voders voice almost loo good to be true ", No restrictions, https://commons.wikimedia.org/w/index.php?curid=43343073

2)
https://deepmind.com/blog/article/wavenet-generative-model-raw-audio

3)
https://colab.research.google.com/notebooks/intro.ipynb

4)
https://simple.wikipedia.org/wiki/Python_(programming_language

5)
https://jupyter.org/

6)
this is not unusual when using a hosted runtime

7)
confusingly we need to call it from bash (with the exclamation

8)
a fancy way of saying option