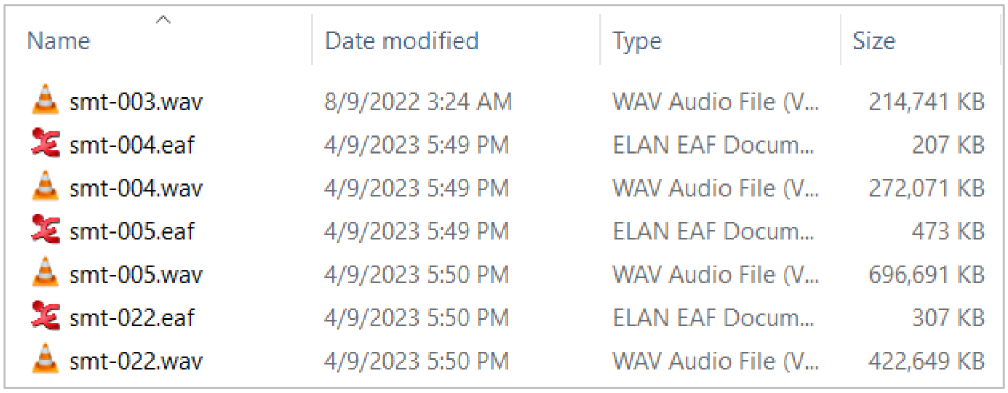
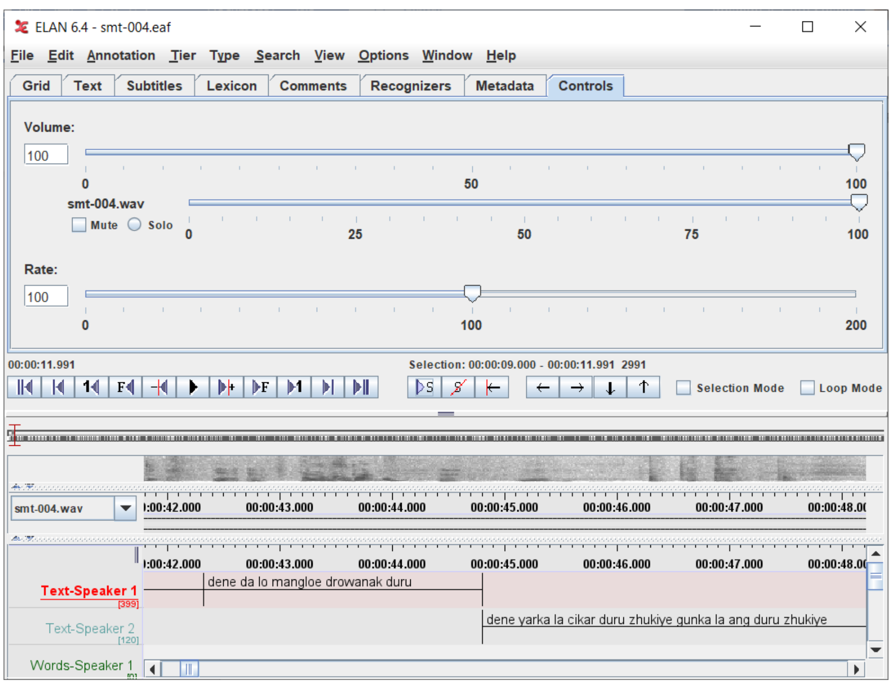
Instructions for Dzardzongke Transcription
Version b03 20230419 2222

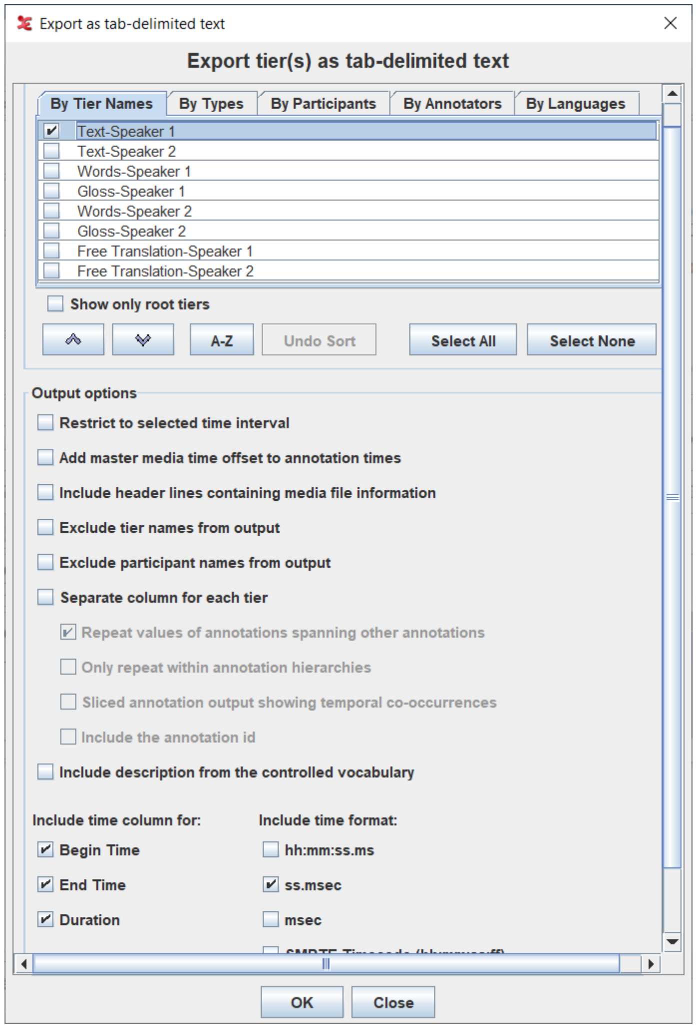
1. We will use three files for training (004, 005, 022) and one file for inference (003).
   You can download them here:
   https://rcweb.dartmouth.edu/homes/f00458c/workshop-nepali-2023/dz-files-20230419.zip



2. Let's open the file `smt-004.eaf`. When you open it you'll see several tiers.



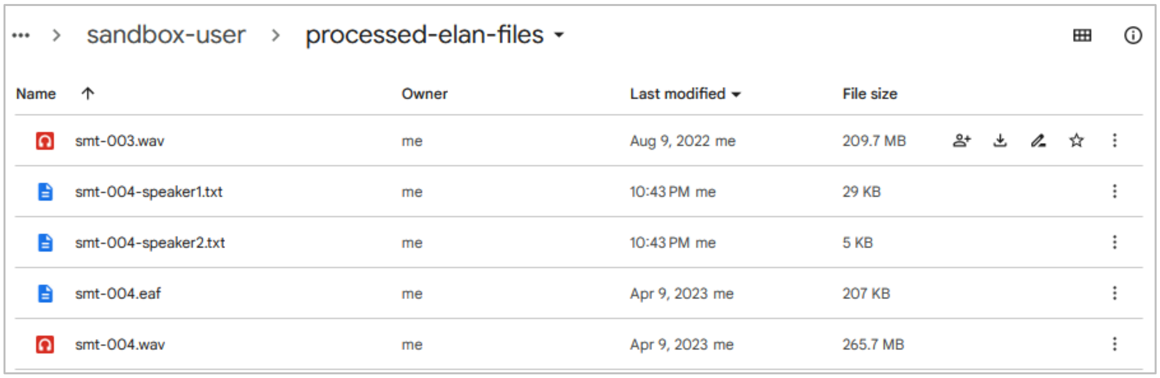
3. We'll export both transcription tiers. Go to *File* > *Export As* > *Tab-delimited Text*. Select one (and only one) of the tiers with transcriptions. At the bottom of the form, select *Begin time*, *End time*, *Duration* and *ss.msec*. (**Don't select *hh:mm:ss.ms***). Click "OK" and save as `smt-004-speaker1.txt`.



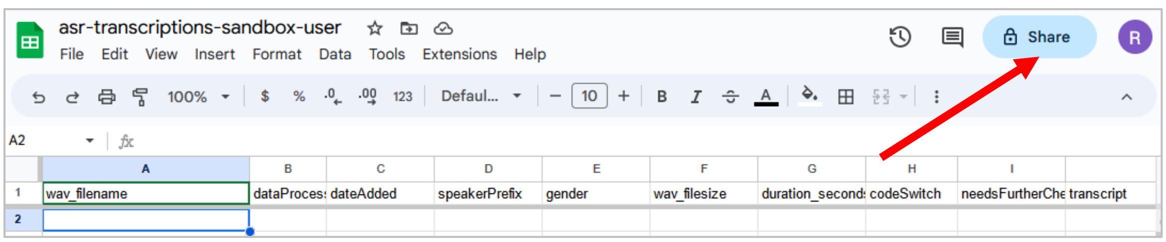
If you open the file, you'll see that it has several tab separated columns. The first one is the name of the tier. The second one is blank (it's the name of the speaker as specified in the tier; there was no speaker metadata in this case). The third one is the beginning of the annotation. The fourth column is the end of the annotation. The fifth column is the duration of the annotation, and the sixth column is the transcription.

Text-Speaker 1		17.03	22.861	5.831	gatsoe droyi yoedro nyishu tsaksum la nama la owa nak
Text-Speaker 1		22.861	25.334	2.473	da danda duencu denga juruk
Text-Speaker 1		25.334	27.789	2.455	da lo mangloe droyik
Text-Speaker 1		29.451	31.516	2.065	kyesa ko chongkhor
Text-Speaker 1		35.115	42.326	7.211	duencu... tsak... ? nyin la tsaksum la nama la owanak danda duer
Text-Speaker 1		42.326	44.803	2.477	dene da lo mangloe drowanak duru
Text-Speaker 1		53.73	57.727	3.997	da yarka gunka duru detik nenying cikar rong la cik ?
Text-Speaker 1		58.373	64.221	5.848	cik sowen balu la pokra la zhen ni mangche duru detik duka duka
Text-Speaker 1		64.221	66.688	2.467	balu mangche duru tsowe detik durue tsowe
Text-Speaker 1		69.923	74.328	4.405	balu la dro na duka duka ra mana...? duka duka dro na pingya tar
Text-Speaker 1		75.832	77.114	1.282	i tshin choedro nang ya? la
Text-Speaker 1		79.01	80.346	1.336	egi puen
Text-Speaker 1		80.346	86.4	6.054	da puen zhi
Text-Speaker 1		86.4	88.476	2.076	bomo puen zhi
Text-Speaker 1		91.952	92.978	1.026	palzang cungshok nak
Text-Speaker 1		94.788	98.051	3.263	da palzang da dene
Text-Speaker 1		98.051	100.148	2.097	da numu cik dang ichi nyi pak?

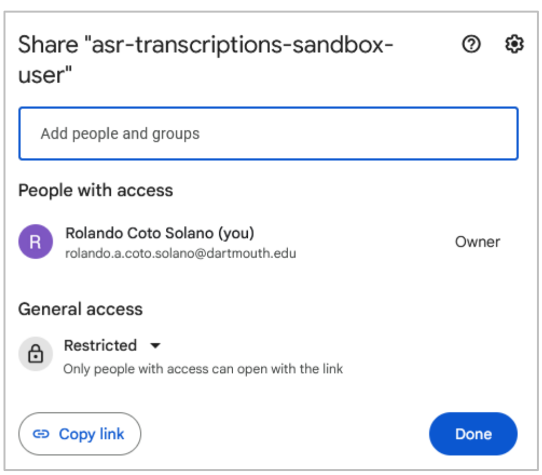
4. Repeat the previous step with the tier for the second speaker (`Text-Speaker 2`). Save this file as `smt-004-speaker2.txt`. Upload the two text files and the audio file into the folder `sandbox-user/processed-elan-files`. Your folder should look like this:



5. Open the `asr-transcriptions-sandbox-user` Google Sheet. (It's in the main `202303-asr-workshop-dz` folder). Click on the "Share button".



This will open up a window that describes the permission to the file. Click on "Copy link". Paste this link somewhere where you can easily retrieve it (e.g. TextEdit). You will need it several times during this process.



6. Go to the notebook `from-elan-to-wav-and-gsheet.ipynb`. Modify the first block of code so that the variables look like this:

destinationSandbox:	`sandbox-user`
installationFolder:	`202303-asr-workshop-dz`
nameTabFile:	`smt-004-speaker1.txt`
speakerCode:	`SP01`
bigWavFile:	`smt-004.wav`
prefixSmallAudioFiles:	`smt-004`
speakerGender:	`f`
urlSandbox	(LINK TO THE SANDBOX SPREADSHEET)

```
[ ] # Fill out the metadata for the file you want to process

    destinationSandbox = "sandbox-user"          # Please type sandbox-user or all-wavs (or one of your user folders)
    installationFolder = "202303-asr-workshop-dz"

    nameTabFile = "smt-004-speaker1.txt"         # The file needs to be in the processed-elan-files folder of your sandbox.
    speakerCode = "SP01"
    bigWavFile = "smt-004.wav"                   # The file needs to be in the processed-elan-files folder of your sandbox.
    prefixSmallAudioFiles = "smt-004"            # Prefix for the split audio files
    speakerGender = "f"                          # Kaldi only has binary m/f. Wav2Vec2 doesn't care

    urlSandbox = "https://docs.google.com/spreadsheets/d/1r97YDkV0Bd3Ohi82MgRhv4lxM2DqtejPP4SRL6JurKw/edit?usp=sharing"
```

(Your URL might be different from the one above).

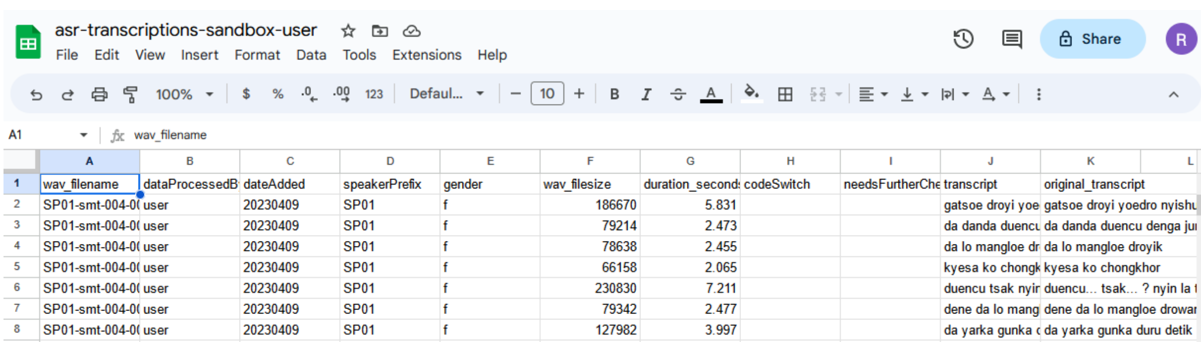
7. Run the first three blocks of code. This will connect the notebook to the Google Drive and have it ready to find the files it needs to process.

8. Run the rest of the blocks. The file should have 388 valid annotations. The process to split the audio files takes some time (approximately 9 minutes). By the end of the notebook you should see something like this:

```
for i in range(0,len(timeStart)):
  with contextlib.closing(wave.open(folderWhereWavsAreStored + filenames[i],'r')) as f:
      frames = f.getnframes()
      rate = f.getframerate()
      duration = frames / float(rate)
  inValues = [filenames[i], destinationSandbox.replace("sandbox-",""), todaysDate, speak
  sandboxRows.loc[len(sandboxRows)] = inValues

set_with_dataframe(sheet, sandboxRows)  # Write all rows to sandbox GSheet
sheet.delete_row(1)                     # Erase the pandas numerical headers
rowsAtEnd = len(sandboxRows.index)

print("Total rows before new data: " + str(rowsAtStart))
print("Total rows after new data:  " + str(rowsAtEnd))
```

```
Total rows before new data: 1
Total rows after new data:  389
```

If you go to the spreadsheet `202303-asr-workshop-dz/asr-transcriptions-sandbox-user`, you will see the transcriptions and their information (e.g. wav file, duration, speaker, etc).



	A	B	C	D	E	F	G	H	I	J	K
1	wav_filename	dataProcessedB	dateAdded	speakerPrefix	gender	wav_filesize	duration_second	codeSwitch	needsFurtherChe	transcript	original_transcript
2	SP01-smt-004-0(	user	20230409	SP01	f	186670	5.831			gatsoe droyi yoe	gatsoe droyi yoedro nyishu
3	SP01-smt-004-0(	user	20230409	SP01	f	79214	2.473			da danda duencu	da danda duencu denga jui
4	SP01-smt-004-0(	user	20230409	SP01	f	78638	2.455			da lo mangloe dr	da lo mangloe droyik
5	SP01-smt-004-0(	user	20230409	SP01	f	66158	2.065			kyesa ko chongk	kyesa ko chongkhor
6	SP01-smt-004-0(	user	20230409	SP01	f	230830	7.211			duencu tsak nyir	duencu... tsak... ? nyin la t
7	SP01-smt-004-0(	user	20230409	SP01	f	79342	2.477			dene da lo mang	dene da lo mangloe drowar
8	SP01-smt-004-0(	user	20230409	SP01	f	127982	3.997			da yarka gunka (	da yarka gunka duru detik

9. Next, repeat steps 6 through 8 for the second tier of the audio file `smt-004`. Save this as `smt-004-speaker2.txt`. Go back to the first block of code in the notebook and modify it as follows:

destinationSandbox:	`sandbox-user`
installationFolder:	`202303-asr-workshop-dz`
nameTabFile:	`smt-004-speaker2.txt`
speakerCode:	`SP02`
bigWavFile:	`smt-004.wav`
prefixSmallAudioFiles:	`smt-004`
speakerGender:	`m`
urlSandbox	(LINK TO THE SANDBOX SPREADSHEET)

```
[1] # Fill out the metadata for the file you want to process

    destinationSandbox = "sandbox-user"          # Please type sandbox-user or all-wavs (or one of your user folders)
    installationFolder = "202303-asr-workshop-dz"

    nameTabFile = "smt-004-speaker2.txt"         # The file needs to be in the processed-elan-files folder of your sandbox.
    speakerCode = "SP02"
    bigWavFile = "smt-004.wav"                   # The file needs to be in the processed-elan-files folder of your sandbox.
    prefixSmallAudioFiles = "smt-004"            # Prefix for the split audio files
    speakerGender = "m"                          # Kaldi only has binary m/f. Wav2Vec2 doesn't care

    urlSandbox = "https://docs.google.com/spreadsheets/d/1SOenjC_NxvY6D5hk3iu7VX7jbAfh0zU4g7ScHKIoCWM/edit?usp=sharing"
```

Rerun all of the blocks of code in the notebook. This will process the files for the second tier and put the transcriptions in the sandbox' Google Sheet. This file only has 18 valid annotations, so it should take a shorter time to process. By the time you have finished, the spreadsheet for the sandbox should have 407 rows.

387	SP01-smt-004-386.wav	user	20230420	SP01	f	130734	4.083			dene palzang de
388	SP01-smt-004-387.wav	user	20230420	SP01	f	138510	4.326			dene bisa drongk
389	SP01-smt-004-388.wav	user	20230420	SP01	f	30062	0.937			da lap ko
390	SP02-smt-004-01.wav	user	20230420	SP02	m	163982	5.122			duru zhukti lo ga
391	SP02-smt-004-02.wav	user	20230420	SP02	m	95790	2.991			duru zhukti lo ga
392	SP02-smt-004-03.wav	user	20230420	SP02	m	148878	4.65			duru ori nama la
393	SP02-smt-004-04.wav	user	20230420	SP02	m	53262	1.662			kyesa ko ta
394	SP02-smt-004-05.wav	user	20230420	SP02	m	115246	3.599			dene duru pura la
395	SP02-smt-004-06.wav	user	20230420	SP02	m	283822	8.867			dene yarka la cik
396	SP02-smt-004-07.wav	user	20230420	SP02	m	103598	3.235			balu la phep na s
397	SP02-smt-004-08.wav	user	20230420	SP02	m	48206	1.504			tshin choedro yo
398	SP02-smt-004-09.wav	user	20230420	SP02	m	81998	2.56			nang ya la puen
399	SP02-smt-004-10.wav	user	20230420	SP02	m	73646	2.299			nyen la ta manga
400	SP02-smt-004-11.wav	user	20230420	SP02	m	37742	1.177			palzang cungsho
401	SP02-smt-004-12.wav	user	20230420	SP02	m	57998	1.81			dene su su duwo
402	SP02-smt-004-13.wav	user	20230420	SP02	m	89134	2.783			dene zhema ta y
403	SP02-smt-004-14.wav	user	20230420	SP02	m	74222	2.317			zhema da drong
404	SP02-smt-004-15.wav	user	20230420	SP02	m	196686	6.144			eh awu thre song
405	SP02-smt-004-16.wav	user	20230420	SP02	m	132814	4.148			awu dand ama ny
406	SP02-smt-004-17.wav	user	20230420	SP02	m	154318	4.82			dene yartung gi k
407	SP02-smt-004-18.wav	user	20230420	SP02	m	110030	3.436			ta mi gyuka

10. Repeat steps 6 through 8 for the other audio and ELAN files: `smt-005` and `smt-022`. The `smt-005`, the person in the first tier is `speaker3`, and the person in the second tier is `speaker2` (they're the same `speaker2` as in the first recording). In `smt-022`, the person in the first tier is `speaker4`, and the speaker in the second tier is again `speaker2`. This is the data that you should enter for each run of the notebook.

    Tier 1, smt-005, 163 valid annotations:
    destinationSandbox: `sandbox-user`
    installationFolder: `202303-asr-workshop-dz`
    nameTabFile: `smt-005-speaker3.txt`
    speakerCode: `SP03`
    bigWavFile: `smt-005.wav`
    prefixSmallAudioFiles: `smt-005`
    speakerGender: `m`
    urlSandbox (LINK TO YOUR SANDBOX SPREADSHEET)

    Tier 2, smt-005, 58 valid annotations:
    destinationSandbox: `sandbox-user`
    installationFolder: `202303-asr-workshop-dz`
    nameTabFile: `smt-005-speaker2.txt`
    speakerCode: `SP02`
    bigWavFile: `smt-005.wav`
    prefixSmallAudioFiles: `smt-005`
    speakerGender: `m`
    urlSandbox (LINK TO YOUR SANDBOX SPREADSHEET)

    Tier 1, smt-022, 431 valid annotations:
    destinationSandbox: `sandbox-user`
    installationFolder: `202303-asr-workshop-dz`
    nameTabFile: `smt-022-speaker4.txt`
    speakerCode: `SP04`
    bigWavFile: `smt-022.wav`
    prefixSmallAudioFiles: `smt-022`
    speakerGender: `m`
    urlSandbox (LINK TO YOUR SANDBOX SPREADSHEET)

    Tier 2, smt-022, 20 valid annotations:
    destinationSandbox: `sandbox-user`
    installationFolder: `202303-asr-workshop-dz`
    nameTabFile: `smt-022-speaker2.txt`
    speakerCode: `SP02`
    bigWavFile: `smt-022.wav`
    prefixSmallAudioFiles: `smt-022`
    speakerGender: `m`
    urlSandbox (LINK TO YOUR SANDBOX SPREADSHEET)

    By the end of this procedure, you should have about 1079 annotations in the sandbox spreadsheet.

11. If you look at the sandbox GSheet closely you'll notice several problems. Are there any blank transcriptions? If so, **add a 1 in column "I" (needsFurtherCheck) of every empty transcription**.

    Are there any of them with code-switching or any other issues? If so, **add a 1 in column "H" (codeSwitch) of every transcription with words in other languages**. For example, the following rows have words in English or Nepali:

    [431 – 435, 450, 454, 580 – 582, 585, 596 – 598]

asr-transcriptions-sandbox-user

File Edit View Insert Format Data Tools Extensions Help

H597 fx 1

	A	B	C	D	E	F	G	H	I	J	K	L
430	SP03-smt-005-023.wav	user	20230420	SP03	m	63406	1.979			Anisya oh	Anisya oh	
431	SP03-smt-005-024.wav	user	20230420	SP03	m	334702	10.457	1		NEP yaha ko khı	[NEP] yaha ko khoi... chavi...	
432	SP03-smt-005-025.wav	user	20230420	SP03	m	99502	3.107	1		NEP tyo chavi cł	[NEP] tyo chavi chaina	
433	SP03-smt-005-026.wav	user	20230420	SP03	m	29646	0.924	1		NEP	[NEP] ...	
434	SP03-smt-005-027.wav	user	20230420	SP03	m	41486	1.294	1		NEP uh	[NEP] uh?	
435	SP03-smt-005-028.wav	user	20230420	SP03	m	94446	2.949	1		NEP jane	[NEP] ...jane	
436	SP03-smt-005-029.wav	user	20230420	SP03	m	57838	1.805			uh thakpa thenke	uh thakpa thenken... tsheme	

12. Go to the notebook `from-gsheet-to-wav2vec2-files.ipynb`. You need to set the installationFolder to `202303-asr-workshop-dz`, and to insert the URL for your sandbox. (Your URL might look different from the one in the example below).

```
destinationSandbox = "sandbox-user"    # Please type sandbox-user or all-wavs
installationFolder = "202303-asr-workshop-dz"

useCodeSwitchedData = 0
useDoutbfulData = 0

percentageTrainSet = 80
percentageValidSet = 10
percentageTestSet = 10

maxWavDuration = 15    # Seconds. This is the longest duration for a file allowed in the dataset. Wav2Vec2's CUDA me
software = "wav2vec2"   # wav2vec2 or ds

urlSandbox = "https://docs.google.com/spreadsheets/d/1SOenjC_NxvY6D5hk3iu7VX7jbAfh0zU4g7ScHKIoCWM/edit?usp=sharing"
```

13. Run all of the code blocks in the notebook. It will ask for permission to access your Google Drive and then split the data into three randomly-assigned sets: training (80%), validation (10%) and testing (10%). At the end there will be three new files in your sandbox:

    `wav2vec2-train.csv`   `wav2vec2-valid.csv`   `wav2vec2-test.csv`

14. Go to the notebook `train-wav2vec2.ipynb`. Modify the first cell so that it has the right installation folder (`202303-asr-workshop-dz`).

```
[50] currentSandbox = "sandbox-user"
     installationFolder = "202303-asr-workshop-dz"

     runId = "01"
     desiredTrainEpochs = 25
```

15. Run all the cells. There is one that has the training in it. It could about 30 minutes, but it could take hours, depending on how much data you have and how many epochs you specified. You'll see results like these. You are okay as long as the validation loss keeps going down. If it bounces and starts going back up, then the model might be "overfitting".

```
trainer.train()
```

[3233/3233 1:07:27, Epoch 60/61]

Step	Training Loss	Validation Loss	Wer
100	7.143900	3.622996	1.000000
200	3.041900	3.048380	1.000000
300	2.972000	2.962219	1.000000
400	2.959100	2.919904	1.000000
500	2.839300	2.655782	1.000000

I performed an experiment to see how the training would behave if it continued. You can see that, by the time you've reached 1200 steps, the validation loss is at 1.74, and from there it starts going up. This indicates that the model is memorizing the patterns in the training set and failing to generalize for what it sees in the validation set. This is called *overfitting*. The WER "word error rate" shows very little gain after 1200 steps, indicating that the model might be stuck. In this particular case you could use the checkpoint 1200. This one shows good learning and a better capacity to generalize than the rest.

1000	0.881600	1.585511	0.905367
1100	0.782600	1.758408	0.895480
1200	0.682200	1.747411	0.864407
1300	0.599400	1.993025	0.887006
1400	0.531500	2.072576	0.867232
1500	0.453500	2.113673	0.878531
1600	0.406800	2.081810	0.870056
1700	0.367900	2.321600	0.872881
1800	0.348700	2.301691	0.861582
1900	0.331600	2.304753	0.851695
2000	0.292400	2.389884	0.810734
2100	0.255400	2.410218	0.838983

2200	0.249300	2.346779	0.846045
2300	0.244400	2.427731	0.840395
2400	0.225300	2.422726	0.844633
2500	0.212100	2.547699	0.840395
2600	0.197100	2.599447	0.844633
2700	0.203400	2.560743	0.824859
2800	0.173300	2.615589	0.836158
2900	0.170200	2.617266	0.837571
3000	0.166800	2.665286	0.824859
3100	0.153800	2.660248	0.836158
3200	0.169700	2.668093	0.831921

16. Once the training is done, the program will run some diagnostics to see how the training did. You can choose which checkpoints to evaluate by looking at this variable. (The number correspond to the steps at which the model will save a "checkpoint").

```
#==============================================================================
# Evaluate checkpoints; calculate their word/character error rates and
# get the predictions for the sentences in the test set.
#==============================================================================

checkpointNums = ["400", "800", "1200"]
```

Here you can see the CER (character error rate) and the WER (word error rate) of each of the checkpoints that were evaluated. (I performed an experiment where I let it run for longer to verify the overfitting. You can see that the checkpoints at 1200 and 1600 steps have similar performance, and after that, there is very little gain).

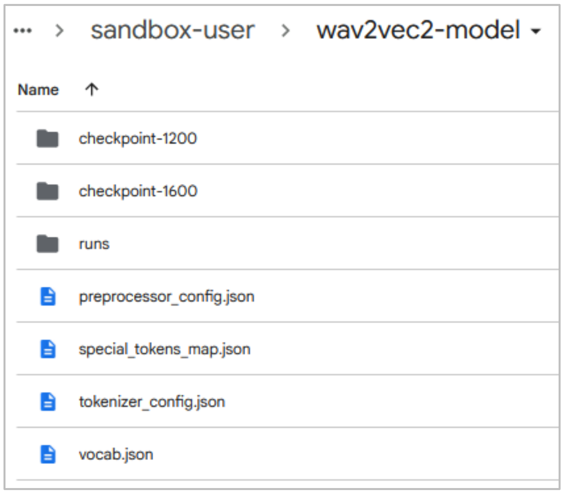
```
01/400 Median CER:      1.0
01/400 Median WER:      1.0

01/800 Median CER:      0.35
01/800 Median WER:      0.866

01/1200 Median CER:     0.279
01/1200 Median WER:     0.75

01/1600 Median CER:     0.276
01/1600 Median WER:     0.714

01/2000 Median CER:     0.292
01/2000 Median WER:     0.721

01/2400 Median CER:     0.265
01/2400 Median WER:     0.667

01/2800 Median CER:     0.259
01/2800 Median WER:     0.75

01/3200 Median CER:     0.262
01/3200 Median WER:     0.707
```

The `saveCheckpoints` variable has a list of the checkpoints that should be saved to your GDrive.

Copy the trained model to your GDrive

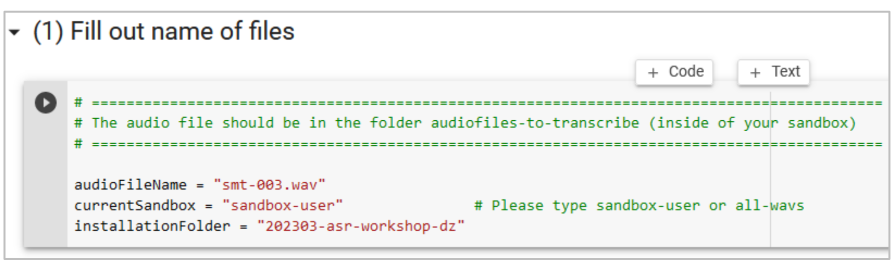
```
[ ]  # Which checkpoint do you want to save?
     saveCheckpoints = ["1200"]
```

17. You can go look at the model files in the folder:
`202303-asr-workshop-dz/sandbox-user/wav2vec2-model`
It has the different checkpoints that were saved. (You should only have the 1200, but you could have others if you have also saved them, like I did below).



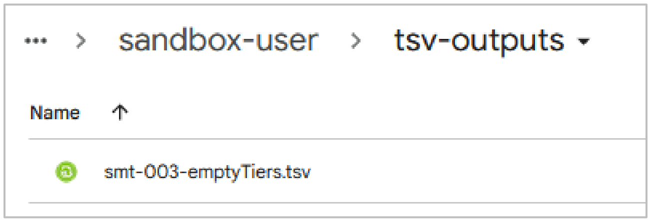
18. Now, let's try to transcribe a completely new file. Choose the file `smt-003.wav` and upload it into the folder `202303-asr-workshop-dz/sandbox-user/audiofiles-to-transcribe`.[1]

19. Next, go to the notebook `inference-split.ipynb`. Please make sure you have the right audio file name (`smt-003.wav`) and the right installation folder (`202303-asr-workshop-dz`). This notebook will try to find segments with speech within the recording.

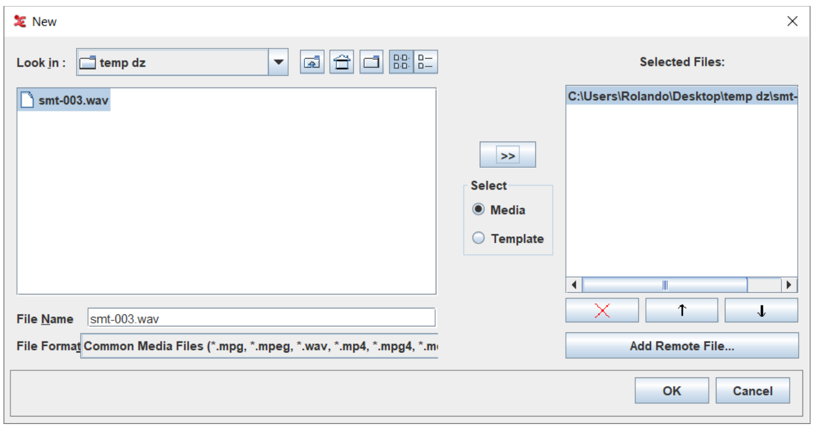


[1] I have already uploaded it to accelerate the exercise.

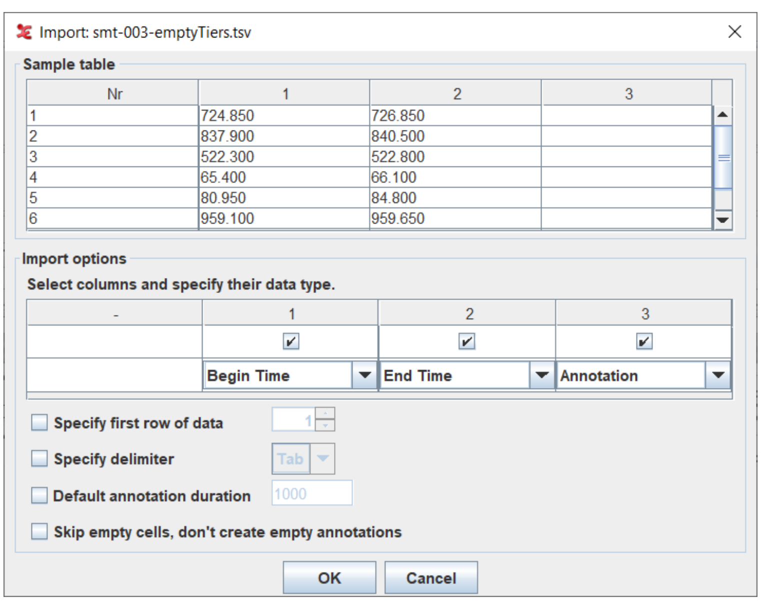
20. Once the notebook has finished running, you will find the file `smt-003-emptyTiers.tsv` in the `tsv-outputs` folder of your sandbox. This will have columns with the start and end times of potential annotations.



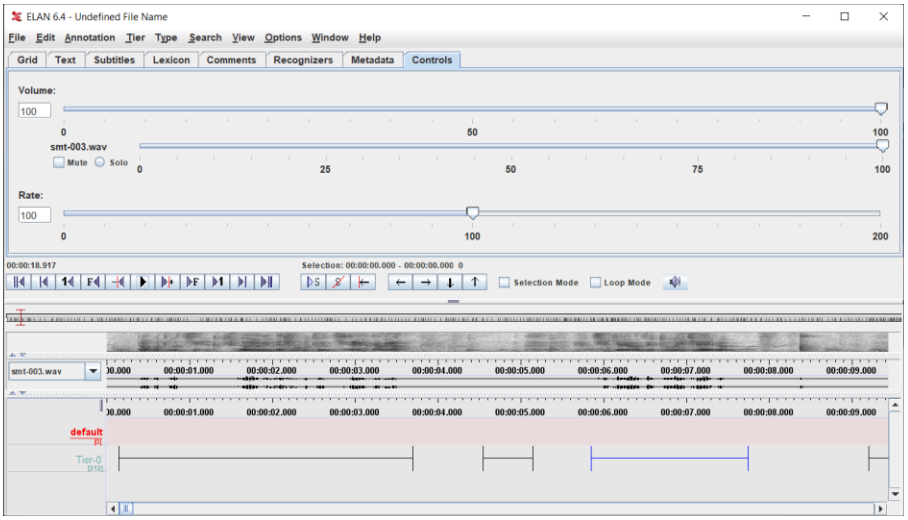
21. Go to ELAN and create a new file with your audio file (`smt-003.wav`).



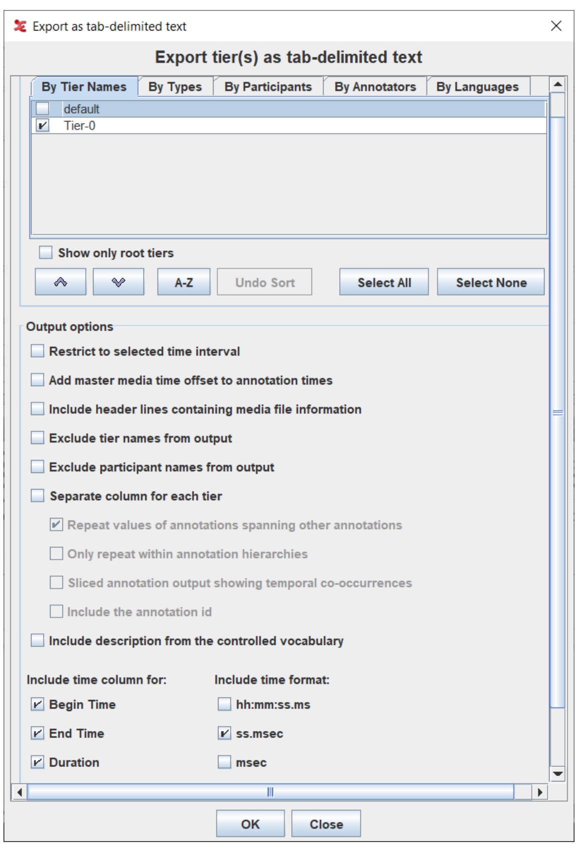
22. Go to *File > Import > CSV / Tab-delimited Text File*. Select the file with the potential annotations (`smt-003-emptyTiers.tsv`). The first column is the *Begin time*. The second column is the *End time*. The third column (which has a single space) is the *Annotation*.



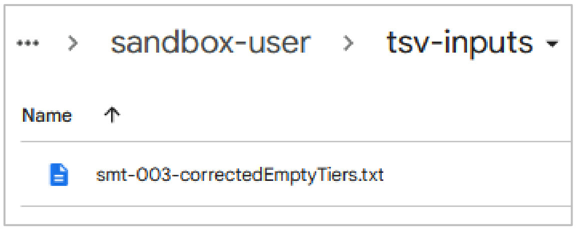
23. You will now see a tier with empty annotations. These are the spaces where the computer found human voice. Check them manually to make sure that the segments are valid. Please make sure you break up any segments longer than 15 seconds.



24. Once you have finished with the manual correction of the empty segments, go to *File* > *Export As* > *Tab-delimited Text*. Select the tier with the corrected empty annotations and export it as `smt-003-correctedEmptyTiers.txt`.

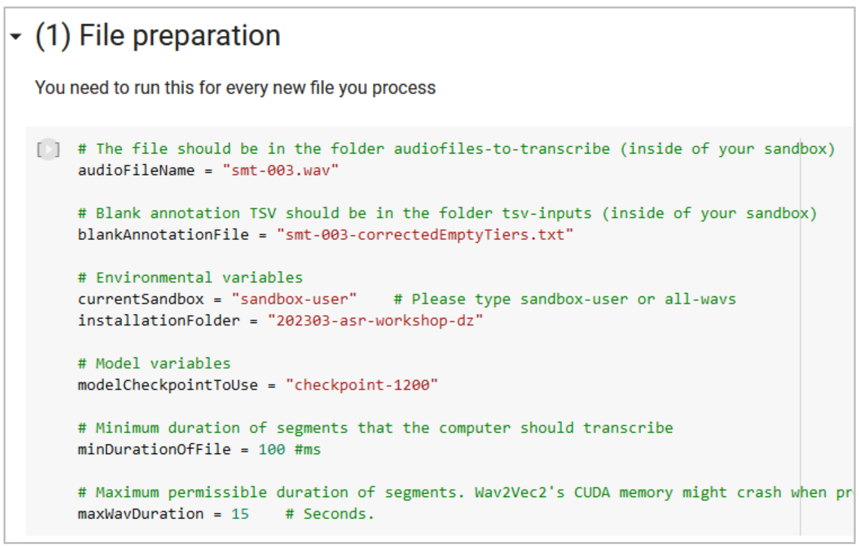


25. Upload the file into the folder `tsv-inputs` of your sandbox.

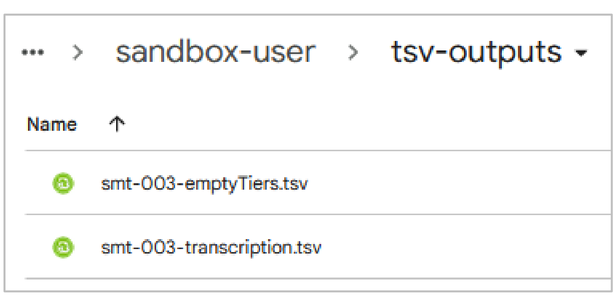


26. Go to the notebook `inference-transcribe-from-blank-elan.ipynb`. You need to change several variables:

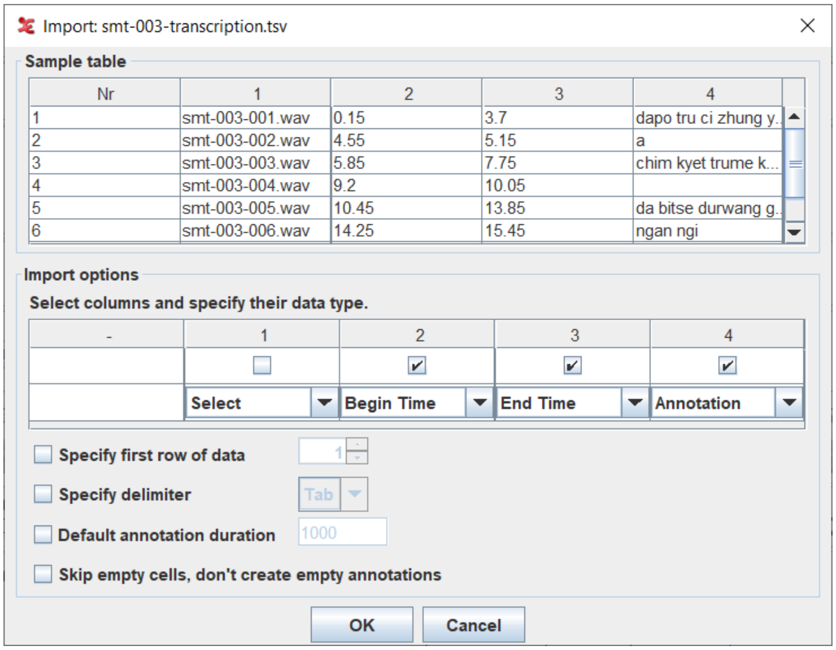
audioFileName:	`smt-003.wav`
blackAnnotationFile	`smt-003-correctedEmptyTiers.txt`
installationFolder	`202303-asr-workshop-dz`
modelCheckpointToUse	`checkpoint-1200`



27. Run all of the code blocks in this notebook. The program will (i) load the times for the annotation, (ii) split the big wav file into smaller pieces, and (iii) transcribe each of the small pieces. (This might take some time to run; please be patient). In the end you will find the file `smt-003-transcription.tsv` in the folder `tsv-outputs` of the sandbox. Go ahead and download it.



28. Go back to the ELAN file for `smt-003`. Go to *File* > *Import* > *CSV / Tab-delimited Text File*. Select the transcription file `smt-003-transcription.tsv`. Please unselect the first column. The second column is the *Begin time*. The third column is the *End time*. The fourth column is the *Annotation*.



29. You will now see one tier with the ASR transcription. You can delete the empty tiers, rename the ASR tier, and move the annotations for different speakers to other tiers. This is the end of the tutorial. You have now trained an ASR model and used it to transcribe your recordings.