Research Design and Pilot Study Report

2. Research Design

2.1 Research goal

Evaluation the effectiveness and accuracy of phrase detection using speech recognition models.

2.2 Research gap

While previous previous Systematic Literature Review showed the evaluations of overall accuracy of speech recognition systems – focusing on metrics like Word Error Rate (WER) and Character Error Rate (CER) - there is limited research on the detection of specific phrases or keywords in audio, especially when these phrases are defined by user. Many apps require efficient identification of some phrases within audio with real-time performance demands, which can be important for example in voice-activated systems.

2.3 Research Questions

1. How to effectively detect user-defined phrases within audio recording?

2. What are the challenges in detecting user-defined phrases with different models of speech recognition?

3. How does the accuracy of user-defined phrases vary across models of different sizes (whisper tiny, base, small, medium, large)?

4. How do different types of phrases (example clear, noisy) affect the success rate of phrase detection?

5. How do emotions and accent in speaking affect the success rate of phrase detection?

2.4 Research Hypotheses

1. User-defined phrases can be effectively detected within audio recordings using modern speech recognition models.

2. The challenges in detecting user-defined phrases depend on the speech recogntion model, where smaller models face bigger difficulties in detecting phrases in noisy recordings.

3. The accuracy of user-defined detection varies between different speech recognition models, larger models outperform smaller models.

4. The detection accuracy of user-defined phrases is affected by the quality of the audio, with clear audio having higher detection success.

5. The detection accuracy of user-defined phrases is affected by emotions of the speaker with neutral and calm way of speaking having higher detection success.

2.5 Research Subjects and Sample

Research subjects - Models

- Whisper Tiny
- Whisper Base
- Whisper Small
- Whisper Medium
- Whisper Large

Sampling Method – Audio recordings (40, 10 for each category):

1. Clean speech (no background noise)
2. Noisy environment (some background noise mixed in)
3. Accent variation (non-native English)
4. Emotion variation (anxious, assertive, encouraging, happy, sad)

Qualification Criteria:

Phrase Presence: contains exactly one occurence of user-defined phrase with

manual annotation.

Length: Between 5 to 15 seconds.

Clarity: The phrase is easy to understand, doesn’t have cut-outs, or loud pops.

2.6 Operationalization – Variables

Variable Type	Variable	Definition
Independent	ASR Model	Studied Model: Whisper: Tiny, Base, Small, Medium, Large
	Audio Condition	Condition of audio, four types: Clean Speech, Noisy Environment, Accent Variation, Emotion variation
Dependent	Phrase Detection Accuracy	Two measures: True Positive Rate (TPR) = (number of correctly detected phrase instances) / (total phrase instances) False Positive Rate (FPR) = (number of detections outside the ground-truth timestamps) / (total non-phrase audio duration)
Confounding	Speaker Gender	Male vs female, could change model performance
	Phrase Position in Clip	Where the phrase appears, later or earlier, could affect model performance
Hidden	Microphone quality	Different microphone settings can make audio quieter or change how clear certain sounds are, but could affect model performance

2.7 Research Methods

Real-Life Experimental Evaluation: One user-defined phrase will be defined and manually marked in 200 real recordings, then run each Whisper's model to try and detect the phrase.

Controlled Simulation: Starting from a clean audio clip I will create new files by adding noise, speeding up, slowind down, and adding echo to the audio, then run each Whisper's model on these to try and detect the phrase.

Prototype Field Test: I will record myself saying phrases once in silence, once with some background noise and feed it into Whisper's model and try to detect the phrase.

2.8 Research Tools

1. Whisper Python Script

Python Script(s) running the OpenAI Whisper Library and gathering the results

2. Audio Annotation Tool

Online website audiomass.co for finding phrases and vscode to edit .csv file with annotations.

3. Audio Processing

Python Script(s) with librosa and soundfile libraries.

4. Audio Recording

Microphone and audio recorder (OBS) with some voice in the background from a computer.

2.9 Expected Results

Quantitative Expected Results:

- Higher TPR is expected from larger Whisper models, compared to smaller ones.
- Lower FPR are expected in clean audio compared to noisy, accented or speed-altered audio.

Qualitative Expected Results:

- Larger Whisper Models are expected to produce fewer misrecognitions.
- Smaller models may struggle with noisy, accented or speed-altered audio.

2.10 Validity Threats

Internal Validity:

The position of the phrase in the audio might affect detection

Microphone quality could influence model performance

Mitigations:

Have phrase in many different positions within the audio

Keep audio recording setup consistent

External Validity:

Testing only one phrase might not generalize

Recordings from only one speaker or accent might limit generalization

Mitigations:

Include many different phrases to test with

Include variations of speakers and accents in audio

Conclusion Validity:

Misintepreting false positives or undetected phrases

Mitigations:

Use objective metrics (TPR, FPR)

2.11 Research Plan

1. Preparation and setup

- Install and configure whisper models with python
- Set up Python Scripts for phrase detection and result extraction
- Set up OBS for recording and VSCode for .csv editing

2. Data Collection and Annotation

- Search for datasets with different audio files that have different audio conditions in them: Clean, noisy, accented, different emotions (200 of them, 50 each)
- Use tools to add noise, speed up, slow down and echo
- Record myself with and without background noise (20 recordings without and 20 with)

- Choose phrases for detection for these files
- Use annotation tool capable of showing audio waves with timings (audiomass.co) to find the phrase and add the data to .csv in code editor (vscode).

3. Model Evaluation

- Run each whisper model on all of these audio samples
- Collect and organize output from these runs, identify true positive and false positives.

4. Analysis and interpretation

- Calculate TPR and FPR for each model and audio conditions.
- Compare performance across models and conditions.

2.12 Publication Goals

What to publish:

- Phrase detection accuracies (TPR and FPR) for different Whisper model sizes
- Comparison tables showing detection accuracy across the four audio conditions

Where to publish:

- IEEE journals
- ACM Digital Library
- ScienceDirect
- ArXiv

3. Pilot Study

3.1 Research subjects

For pilot study, limited subset of research subjects was selected

Models:

- Whisper Tiny
- Whisper Large

Dataset Audio Files:

- Clean speech: 3 audio files
- Noisy environment: 3 audio files
- Accent variation: 3 audio files
- Emotions variation: 3 audio files

Self-recorded audio:

- without background noise: 2 recordings
- with background noise: 2 recordings

Each file above will have one modification done to it:

- Added noise: 16 files in total
- Slowed down: 16 files in total
- Speed up: 16 files in total
- Added Echo: 16 files in total

Target phrases:

Cheese, reach, bags, worried, likes, try

Criteria:

- Each recording contains one occurence of the phrase.
- Each recording is between 5 to 10 seconds in length.
- The phrase is clearly spoken in all recordings.
- Manual annotation was performed to mark the timestamp of phrases in audio.

3.2 Study Execution

First i downloaded datasets for all 4 categories:

- Noisy and clean sets - https://datashare.ed.ac.uk/handle/10283/2791
- Accent set - https://www.kaggle.com/datasets/rtatman/speech-accent-archive?resource=download
- Emotion set - https://www.kaggle.com/datasets/tli725/jl-corpus

After downloading these I selected some of the files for each category as described in 3.1.

I recorded myself using OBS with and without background noise, with 2 different texts, 2 with added background noise, 2 clean, resulting in 4 files in total.

Then for all of these audio files I made additional 4 files, each with one modification, using a simple python script that uses librosa and soundfile libraries to add effects.

```python
import librosa
import librosa.effects as effects
import soundfile as sf
import numpy as np
import os

def add_noise(audio_path, output_path, noise_level=0.05):
    y, sr = librosa.load(audio_path)
    noise = np.random.normal(0, noise_level, y.shape)
    y_noisy = y + noise
    sf.write(output_path, y_noisy, sr)

def add_echo(audio_path, output_path, delay=0.1, decay=0.3):
    y, sr = librosa.load(audio_path)
    delay_samples = int(delay * sr)
    echo = np.zeros_like(y)
    echo[delay_samples:] = y[:-delay_samples] * decay
    y_echo = y + echo
    sf.write(output_path, y_echo, sr)

def speed_up(audio_path, output_path, rate=1.5):
    y, sr = librosa.load(audio_path)
    y_fast = effects.time_stretch(y, rate=rate)
    sf.write(output_path, y_fast, sr)

def slow_down(audio_path, output_path, rate=0.75):
    y, sr = librosa.load(audio_path)
    y_slow = effects.time_stretch(y, rate=rate)
    sf.write(output_path, y_slow, sr)

path = "D:/SoundFilter/pilot_study/noisy_testset_wav/p257_023.wav"

filename = os.path.basename(path)
name_without_ext = os.path.splitext(filename)[0]
```

Then i annotated selected phrase for each file in visual studio code by editing .csv file. For checking the timings for phrases I used website called: audiomass.co.

It resulted in the following .csv file:

	column 1	column 2	column 3	column 4
1	audio_file	phrase	start_time	end_time
2	afrikaans1.mp3	cheese	7.9	8.3
3	french1.mp3	cheese	11.3	11.8
4	german1.mp3	cheese	10.8	11.2
5	afrikaans1_slow.wav	cheese	10.7	11.2
6	french1_slow.wav	cheese	15.1	15.8
7	german1_slow.wav	cheese	14.5	15.2
8	afrikaans1_echo.wav	cheese	8.02	8.4
9	french1_echo	cheese	11.3	11.7
10	german1_echo.wav	cheese	10.8	11.3
11	afrikaans1_noise.wav	cheese	7.9	8.3
12	french1_noise.wav	cheese	11.3	11.8
13	german1_noise.wav	cheese	10.8	11.3
14	afrikaans1_fast.wav	cheese	5.3	5.7
15	french1_fast.wav	cheese	7.7	8.1
16	german1_fast.wav	cheese	7.4	7.8
17	p232_003_c.wav	cheese	4.348	4.8
18	p232_005_c.wav	bags	2.5	3.0
19	p232_011_c.wav	reach	2.075	2.5
20	p232_003_echo_c.wav	cheese	4.348	4.8
21	p232_005_echo_c.wav	bags	2.5	3.0
22	p232_011_echo_c.wav	reach	2.075	2.5
23	p232_003_noise_c.wav	cheese	4.348	4.8
24	p232_005_noise_c.wav	bags	2.5	3.0
25	p232_011_noise_c.wav	reach	2.075	2.5
26	p232_003_slow_c.wav	cheese	5.745	6.5
27	p232_005_slow_c.wav	bags	3.4	4.08
28	p232_011_slow_c.wav	reach	2.75	3.3

Of manually labeled timings of chosen phrases.

After labeling the files i made a script that uses Whisper's tiny and large models to try and find target phrases and save the results in separate .csv file.

Part of console output of running the script:

```
Processing 1-noise.mp4 with large model...
Transcript:  I bought some fresh cheese at the market yesterday, it tastes amazing with crackers and a glass of wine.
  Phrase 'cheese' in 1-noise.mp4: 1 detections
Processing 1-noise_echo.wav with large model...
Transcript:  I bought some fresh cheese at the market yesterday, it tastes amazing with crackers and a glass of wine.
  Phrase 'cheese' in 1-noise_echo.wav: 1 detections
Processing 1-noise_fast.wav with large model...
Transcript:  Thank you so much, Jason. We hope you have a nice and easy progress in the next one.
  Phrase 'cheese' in 1-noise_fast.wav: 0 detections
Processing 1-noise_noise.wav with large model...
Transcript:  I note some bad things that were parted yesterday in place of the nation's progress in the last moment.
  Phrase 'cheese' in 1-noise_noise.wav: 0 detections
Processing 1-noise_slow.wav with large model...
Transcript:  I love some french cheese that I bought yesterday. It tastes just amazing. My crackers and my glass of wine
```

3.3 Results

The resulting .csv files from trying to simply detect the phrase with Whisper models Tiny and Large:

	column 1	column 2	column 3	column 4	column 5	column 6	column 7
1	model	audio_file	phrase	detected_start	detected_end	actual_start	actual_end
2	tiny	1-noise.mp4	cheese	2.4	2.78	2.4	2.7
3	tiny	1-noise_echo.mp4	cheese	2.4	2.8	2.4	2.7
4	tiny	1-noise_fast.mp4	cheese			1.6	1.9
5	tiny	1-noise_noise.mp4	cheese			2.4	2.7
6	tiny	1-noise_slow.mp4	cheese			3.2	3.8
7	tiny	1.mp4	cheese	2.12	2.54	1.95	2.55
8	tiny	1_echo.mp4	cheese	2.12	2.58	1.95	2.55
9	tiny	1_fast.mp4	cheese	1.44	1.74	1.3	1.7
10	tiny	1_noise.mp4	cheese			1.95	2.55
11	tiny	1_slow.mp4	cheese	2.88	3.42	2.7	3.4
12	tiny	2-noise.mp4	cheese			4.5	5.2
13	tiny	2-noise_echo.mp4	cheese			4.5	5.2
14	tiny	2-noise_fast.mp4	cheese			3.0	3.3
15	tiny	2-noise_noise.mp4	cheese			4.5	5.2
16	tiny	2-noise_slow.mp4	cheese			6.0	6.8
17	tiny	2.mp4	cheese	4.66	5.06	4.7	5.02
18	tiny	2_echo.mp4	cheese	4.56	5.04	4.7	5.02
19	tiny	2_fast.mp4	cheese			3.04	3.36
20	tiny	2_noise.mp4	cheese			4.7	5.02
21	tiny	2_slow.mp4	cheese			6.2	6.7
22	tiny	afrikaans1.mp3	cheese	8.0	8.34	7.9	8.3
23	tiny	afrikaans1_echo.wav	cheese	8.02	8.32	8.02	8.4
24	tiny	afrikaans1_fast.wav	cheese			5.3	5.7
25	tiny	afrikaans1_noise.wav	cheese			7.9	8.3
26	tiny	afrikaans1_slow.wav	cheese	10.68	11.1	10.7	11.2
27	tiny	female1_anxious_14b_1.wav	worried	1.32	1.66	1.3	1.65
28	tiny	female1_anxious_14b_1_echo.wav	worried	1.26	1.66	1.3	1.65

And additionally a separate .csv file that includes transcripts:

	column 1	column 2	column 3	column 4	column 5	column 6	column 7	
1	model	audio_file	phrase	detected_start	detected_end	actual_start	actual_end	transcript
2	tiny	1-noise.mp4	cheese	2.4	2.78	2.4	2.7	I bought some fresh cheese at the park to be a swimmer, it tasted amazing with
3	tiny	1-noise_echo.mp4	cheese	2.42	2.78	2.4	2.7	I am not some fresh cheese, i can eat it some days, it will crack it some less if i s
4	tiny	1-noise_fast.mp4	cheese			1.6	1.9	어서 Truj Qui 왔
5	tiny	1-noise_noise.mp4	cheese			2.4	2.7	
6	tiny	1-noise_slow.mp4	cheese			3.2	3.8	Well over expectation. Cochbral�
7	tiny	1.mp4	cheese	2.12	2.54	1.95	2.55	I bought some fresh cheese at the market yesterday, it tasted amazing with cra
8	tiny	1_echo.mp4	cheese	2.12	2.58	1.95	2.55	I bought some fresh cheese for the market yesterday and it tasted amazing with
9	tiny	1_fast.mp4	cheese	1.44	1.74	1.3	1.7	I bought some fresh cheese and the market yesterday. It tastes as amazing with
10	tiny	1_noise.mp4	cheese			1.95	2.55	
11	tiny	1_slow.mp4	cheese	2.88	3.42	2.7	3.4	I brought some fresh cheese at the market yesterday, it tasted amazing with cra
12	tiny	2-noise.mp4	cheese			4.5	5.2	There are kept BulletScore flights, tournaments start from 3. Please come up to
13	tiny	2-noise_echo.mp4	cheese			4.5	5.2	the research that we result in a lot of this did get the appropriate options
14	tiny	2-noise_fast.mp4	cheese			3.0	3.3	Yeah, this is possibly a huge
15	tiny	2-noise_noise.mp4	cheese			4.5	5.2	
16	tiny	2-noise_slow.mp4	cheese			6.0	6.8	This is the cause for free. We dance. We are free. It's not fair enough like you.
17	tiny	2.mp4	cheese	4.66	5.06	4.7	5.02	The recipe calls for free ingredients, milk, flour and cheese mix them together a
18	tiny	2_echo.mp4	cheese	4.56	5.04	4.7	5.02	There are a set of calls for three ingredients, milk, flour and cheese mixed toget
19	tiny	2_fast.mp4	cheese			3.04	3.36	If possible because pre-engrian Snake, function on the intereges ready, one ma
20	tiny	2_noise.mp4	cheese			4.7	5.02	
21	tiny	2_slow.mp4	cheese			6.2	6.7	Fl!</p!</b?! Just shaped사를 Ka 1mm fast." Kap cabbage sientes rectangular te
22	tiny	afrikaans1.mp3	cheese	8.0	8.34	7.9	8.3	Please call Stala, ask her to bring these things with her from the store 6 spoons
23	tiny	afrikaans1_echo.wav	cheese	8.02	8.32	8.02	8.4	Please call Stella, ask her to bring these things with her from those six points of
24	tiny	afrikaans1_fast.wav	cheese			5.3	5.7	This all set enable to bring boxes enjoyment , trifle G prop words, also welcome
25	tiny	afrikaans1_noise.wav	cheese			7.9	8.3	This__scaler worshipped the thing we created here by this blue fox earbehaved
26	tiny	afrikaans1_slow.wav	cheese	10.68	11.1	10.7	11.2	These goals teller are they to bring these things with them from those six spoo
27	tiny	female1_anxious_14b_1.wav	worried	1.32	1.66	1.3	1.65	I am so worried about my exam results.
28	tiny	female1_anxious_14b_1_echo.wav	worried	1.26	1.66	1.3	1.65	I am so worried about my exam results.
29	tiny	female1_anxious_14b_1_fast.wav	worried			0.8	1.174	I have sobered about my exam results.
30	tiny	female1_anxious_14b_1_noise.wav	worried			1.3	1.65	
31	tiny	female1_anxious_14b_1_slow.wav	worried	1.72	2.18	1.68	2.34	I am so worried about my exerosults.
32	tiny	female1_assertive_14a_2.wav	try	0.54	1.1	0.6	1.07	Please try to do your exercises next time.
33	tiny	female1_assertive_14a_2_echo.wav	try	0.58	1.12	0.6	1.07	Please try to do your exercises next time.
34	tiny	female1_assertive_14a_2_fast.wav	try	0.46	0.74	0.488	0.8	Please try to do your exercises next time.
35	tiny	female1_assertive_14a_2_noise.wav	try			0.6	1.07	
36	tiny	female1_assertive_14a_2_slow.wav	try	0.78	1.5	1.0	1.6	Please try to do your exercises next time.
37	tiny	female1_encouraging_4b_2.wav	likes	0.7	1.2	0.7	1.16	Taylor likes Jude Asian food.
38	tiny	female1_encouraging_4b_2_echo.wav	likes	0.62	1.1	0.7	1.16	Taylor likes Jude Asian food.
39	tiny	female1_encouraging_4b_2_fast.wav	likes	0.46	0.8	0.65	0.79	Taylor likes to eat a food.
40	tiny	female1_encouraging_4b_2_noise.wav	likes			0.7	1.16	
41	tiny	female1_encouraging_4b_2_slow.wav	likes	0.82	1.58	1.0	1.55	Taylor likes Jude Asian food.
42	tiny	french1.mp3	cheese	11.36	11.84	11.3	11.8	Please co-stella, as her to bring these things with her from the star, 6 spoons of
43	tiny	french1_echo	cheese	11.38	11.84	11.3	11.7	Please call Stella, ask her to bring these things with her from the star, six spoon
44	tiny	french1_fast.wav	cheese	7.62	7.9	7.7	8.1	Please go to the lab as her to bring these things with her for the star 6 months
45	tiny	french1_noise.wav	cheese			11.3	11.8	Magically we clean the wood we can see that we were going out of the same tä
46	tiny	french1_slow.wav	cheese	15.14	15.74	15.1	15.8	Please co-stella, ask her to bring these things with her from the star, the experi

With these .csv files I calculated the metrics and I will display them with .csv files and diagrams.

Metrics Calculation

False Positive:

We will calculate how much of overlap there is between the detected phrase and the actual phrase. For example if the phrase “cheese” appears at 2.92 - 3.48 seconds and the actual phrase occurs at 3.2 - 3.8 seconds, then the overlap is 0.28 (3.48 - 3.2), if we assume the threshold of **50%**, 0.28 is less than 50% of 0.6, so it’s a **False Positive**.

The threshold was set at: **15%**

True Positive:

The overlap has to be greater or equal to **15%** of actual phrase duration.

Note: True Positive and False Positive are both **False** if phrase was not detected by model at all.

FPR & TPR (False Positive Rate & True Positive Rate):

TPR = True positives / Total Phrases

FPR = False positives / Total Phrases

Phrase position classification:

If phrase is before 3 seconds it's marked as **beginning**, less than 7 seconds is **middle** and else is **end**.

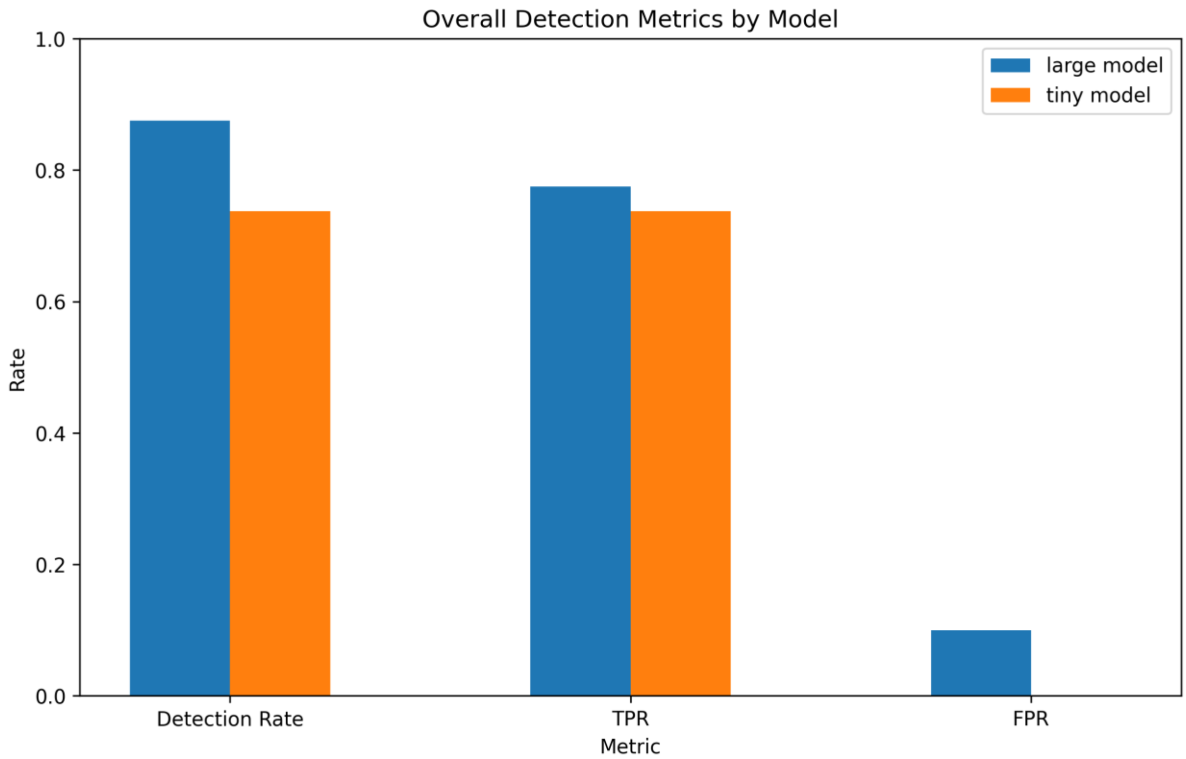
Overall Results:

	column 1	column 2	column 3	column 4	column 5	column 6	column 7	column 8	column 9	column 10
1	model	category	subcategory	total_phrases	true_positives	false_positives	detections_made	tpr	fpr	detection_rate
2	large	all	all	80	62	8	70	0.775	0.1	0.875
3	large	condition	echo	14	13	1	14	0.9285714285714286	0.07142857142857142	1.0
4	large	condition	fast	14	10	4	14	0.7142857142857143	0.2857142857142857	1.0
5	large	condition	noise	24	14	0	14	0.5833333333333334	0.0	0.5833333333333334
6	large	condition	original	14	13	1	14	0.9285714285714286	0.07142857142857142	1.0
7	large	condition	slow	14	12	2	14	0.8571428571428571	0.14285714285714285	1.0
8	large	speech_category	accent	15	11	3	14	0.7333333333333333	0.2	0.9333333333333333
9	large	speech_category	emotion	15	9	3	12	0.6	0.2	0.8
10	large	speech_category	standard	50	42	2	44	0.84	0.04	0.88
11	large	phrase	bags	10	10	0	10	1.0	0.0	1.0
12	large	phrase	cheese	45	33	5	38	0.7333333333333333	0.1111111111111111	0.8444444444444444
13	large	phrase	likes	5	3	1	4	0.6	0.2	0.8
14	large	phrase	reach	10	10	0	10	1.0	0.0	1.0
15	large	phrase	try	5	2	2	4	0.4	0.4	0.8
16	large	phrase	worried	5	4	0	4	0.8	0.0	0.8
17	large	phrase_position	beginning	44	35	3	38	0.7954545454545454	0.06818181818181818	0.8636363636363636
18	large	phrase_position	end	14	10	3	13	0.7142857142857143	0.21428571428571427	0.9285714285714286
19	large	phrase_position	middle	22	17	2	19	0.7727272727272727	0.09090909090909091	0.8636363636363636
20	tiny	all	all	80	59	0	59	0.7375	0.0	0.7375
21	tiny	condition	echo	14	14	0	14	1.0	0.0	1.0
22	tiny	condition	fast	14	10	0	10	0.7142857142857143	0.0	0.7142857142857143
23	tiny	condition	noise	24	8	0	8	0.3333333333333333	0.0	0.3333333333333333
24	tiny	condition	original	14	14	0	14	1.0	0.0	1.0
25	tiny	condition	slow	14	13	0	13	0.9285714285714286	0.0	0.9285714285714286
26	tiny	speech_category	accent	15	11	0	11	0.7333333333333333	0.0	0.7333333333333333
27	tiny	speech_category	emotion	15	11	0	11	0.7333333333333333	0.0	0.7333333333333333
28	tiny	speech_category	standard	50	37	0	37	0.74	0.0	0.74
29	tiny	phrase	bags	10	10	0	10	1.0	0.0	1.0
30	tiny	phrase	cheese	45	28	0	28	0.6222222222222222	0.0	0.6222222222222222
31	tiny	phrase	likes	5	4	0	4	0.8	0.0	0.8
32	tiny	phrase	reach	10	10	0	10	1.0	0.0	1.0
33	tiny	phrase	try	5	4	0	4	0.8	0.0	0.8
34	tiny	phrase	worried	5	3	0	3	0.6	0.0	0.6
35	tiny	phrase_position	beginning	44	36	0	36	0.8181818181818182	0.0	0.8181818181818182
36	tiny	phrase_position	end	14	11	0	11	0.7857142857142857	0.0	0.7857142857142857
37	tiny	phrase_position	middle	22	12	0	12	0.5454545454545454	0.0	0.5454545454545454

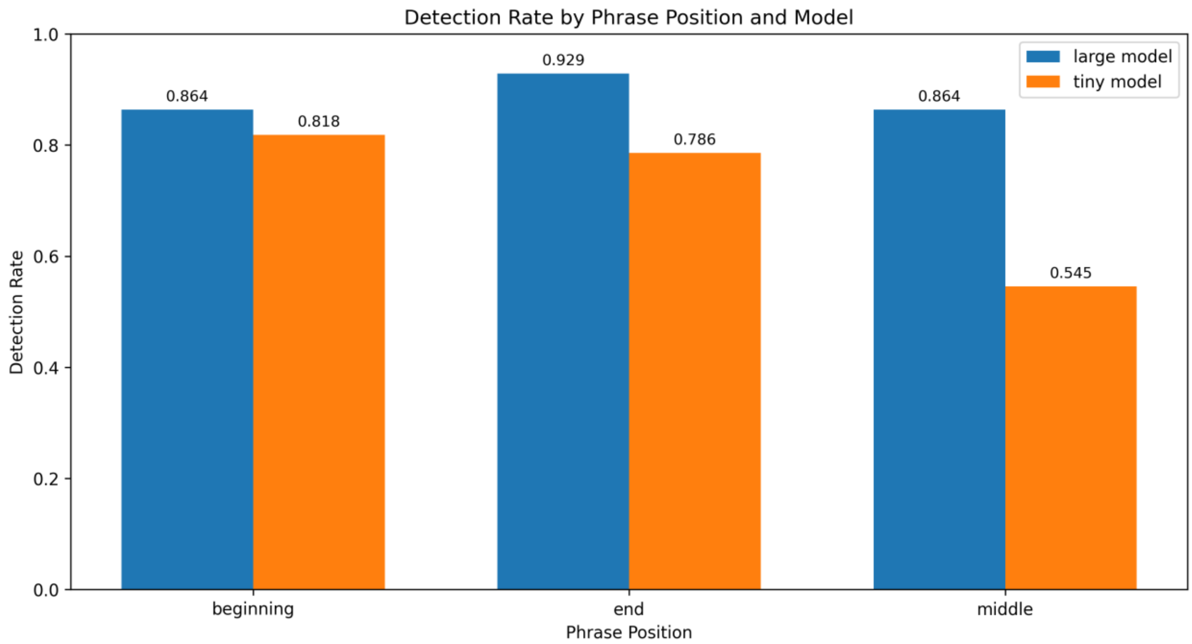
Detailed Results:

	column 1	column 2	column 3	column 4	column 5	column 6	column 7	column 8	column 9	column 10	column 11	column 12	column 13
1	model	audio_file	phrase	detected_start	detected_end	actual_start	actual_end	condition	speech_category	phrase_position	true_positive	false_positive	detection_made
2	tiny	1-noise.mp4	cheese	2.4	2.78	2.4	2.7	noise	standard	beginning	True	False	True
3	tiny	1-noise_echo.mp4	cheese	2.4	2.8	2.4	2.7	noise	standard	beginning	True	False	True
4	tiny	1-noise_fast.mp4	cheese			1.6	1.9	noise	standard	beginning	False	False	False
5	tiny	1-noise_noise.mp4	cheese			2.4	2.7	noise	standard	beginning	False	False	False
6	tiny	1-noise_slow.mp4	cheese			3.2	3.8	noise	standard	middle	False	False	False
7	tiny	1.mp4	cheese	2.12	2.54	1.95	2.55	original	standard	beginning	True	False	True
8	tiny	1_echo.mp4	cheese	2.12	2.58	1.95	2.55	echo	standard	beginning	True	False	True
9	tiny	1_fast.mp4	cheese	1.44	1.74	1.3	1.7	fast	standard	beginning	True	False	True
10	tiny	1_noise.mp4	cheese			1.95	2.55	noise	standard	beginning	False	False	False
11	tiny	1_slow.mp4	cheese	2.88	3.42	2.7	3.4	slow	standard	beginning	True	False	True
12	tiny	2-noise.mp4	cheese			4.5	5.2	noise	standard	middle	False	False	False
13	tiny	2-noise_echo.mp4	cheese			4.5	5.2	noise	standard	middle	False	False	False
14	tiny	2-noise_fast.mp4	cheese			3.0	3.3	noise	standard	middle	False	False	False
15	tiny	2-noise_noise.mp4	cheese			4.5	5.2	noise	standard	middle	False	False	False
16	tiny	2-noise_slow.mp4	cheese			6.0	6.8	noise	standard	middle	False	False	False
17	tiny	2.mp4	cheese	4.66	5.06	4.7	5.02	original	standard	middle	True	False	True
18	tiny	2_echo.mp4	cheese	4.56	5.04	4.7	5.02	echo	standard	middle	True	False	True
19	tiny	2_fast.mp4	cheese			3.04	3.36	fast	standard	middle	False	False	False
20	tiny	2_noise.mp4	cheese			4.7	5.02	noise	standard	middle	False	False	False
21	tiny	2_slow.mp4	cheese			6.2	6.7	slow	standard	middle	False	False	False
22	tiny	afrikaans1.mp3	cheese	8.0	8.34	7.9	8.3	original	accent	end	True	False	True
23	tiny	afrikaans1_echo.wav	cheese	8.02	8.32	8.02	8.4	echo	accent	end	True	False	True
24	tiny	afrikaans1_fast.wav	cheese			5.3	5.7	fast	accent	middle	False	False	False
25	tiny	afrikaans1_noise.wav	cheese			7.9	8.3	noise	accent	end	False	False	False
26	tiny	afrikaans1_slow.wav	cheese	10.68	11.1	10.7	11.2	slow	accent	end	True	False	True
27	tiny	female1_anxious_14b_1.wav	worried	1.32	1.66	1.3	1.65	original	emotion	beginning	True	False	True
28	tiny	female1_anxious_14b_1_echo.wav	worried	1.26	1.66	1.3	1.65	echo	emotion	beginning	True	False	True
29	tiny	female1_anxious_14b_1_fast.wav	worried			0.8	1.174	fast	emotion	beginning	False	False	False
30	tiny	female1_anxious_14b_1_noise.wav	worried			1.3	1.65	noise	emotion	beginning	False	False	False
31	tiny	female1_anxious_14b_1_slow.wav	worried	1.72	2.18	1.68	2.34	slow	emotion	beginning	True	False	True
32	tiny	female1_assertive_14a_2.wav	try	0.54	1.1	0.6	1.07	original	emotion	beginning	True	False	True
33	tiny	female1_assertive_14a_2_echo.wav	try	0.58	1.12	0.6	1.07	echo	emotion	beginning	True	False	True
34	tiny	female1_assertive_14a_2_fast.wav	try	0.46	0.74	0.488	0.8	fast	emotion	beginning	True	False	True
35	tiny	female1_assertive_14a_2_noise.wav	try			0.6	1.07	noise	emotion	beginning	False	False	False
36	tiny	female1_assertive_14a_2_slow.wav	try	0.78	1.5	1.0	1.6	slow	emotion	beginning	True	False	True
37	tiny	female1_encouraging_4b_2.wav	likes	0.7	1.2	0.7	1.16	original	emotion	beginning	True	False	True
38	tiny	female1_encouraging_4b_2_echo.wav	likes	0.62	1.1	0.7	1.16	echo	emotion	beginning	True	False	True
39	tiny	female1_encouraging_4b_2_fast.wav	likes	0.46	0.8	0.65	0.79	fast	emotion	beginning	True	False	True
40	tiny	female1_encouraging_4b_2_noise.wav	likes			0.7	1.16	noise	emotion	beginning	False	False	False
41	tiny	female1_encouraging_4b_2_slow.wav	likes	0.82	1.58	1.0	1.55	slow	emotion	beginning	True	False	True

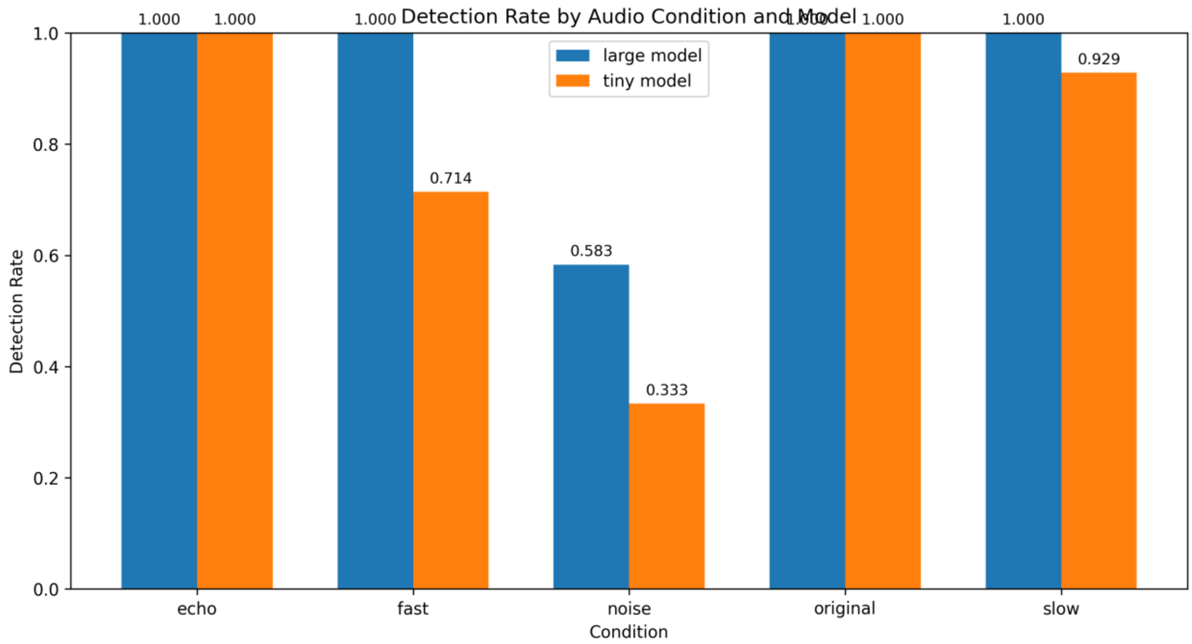
Overall Metrics Comparison & Other Diagrams:



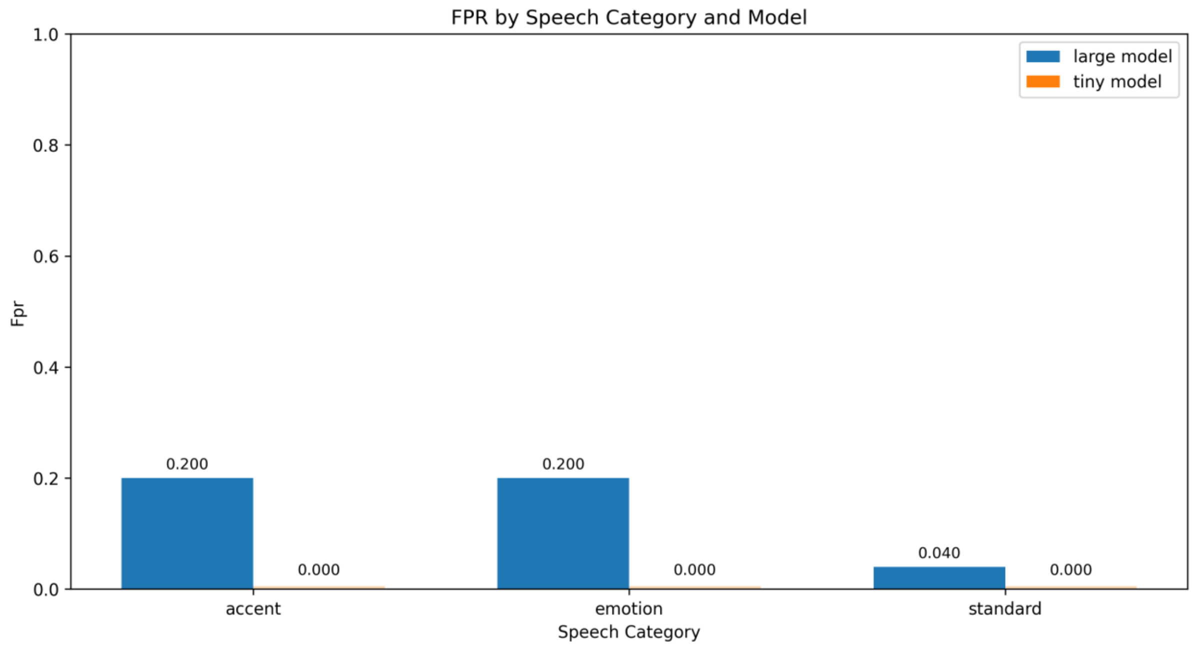

Detection Rate by Speech Category and Model
Detection Rate
Speech Category
accent
emotion
standard
large model
tiny model
0.933
0.733
0.800
0.733
0.880
0.740
0.0
0.2
0.4
0.6
0.8
1.0

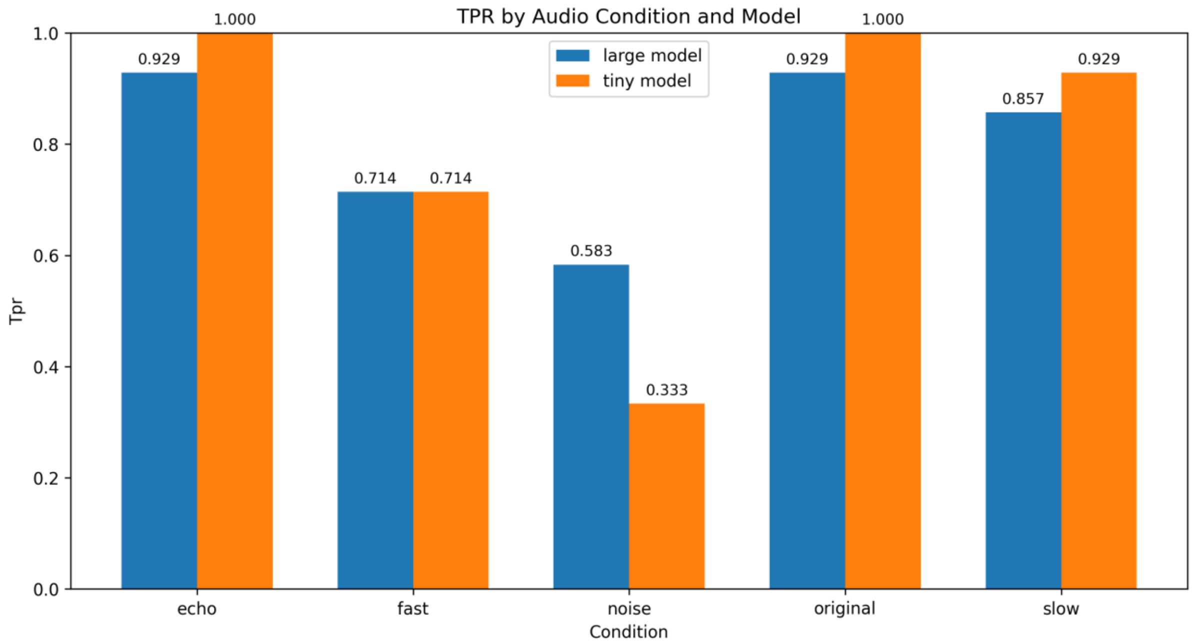


Detection Rate by Phrase Position and Model
large model
tiny model
0.864
0.818
0.929
0.786
0.864
0.545
Detection Rate
0.0
0.2
0.4
0.6
0.8
1.0
beginning
end
middle
Phrase Position



Detection Rate by Audio Condition and Model
large model
tiny model
1.000
1.000
1.000
0.714
0.583
0.333
1.000
1.000
1.000
0.929
Detection Rate
0.0
0.2
0.4
0.6
0.8
1.0
echo
fast
noise
original
slow
Condition





Quality Evaluation

FPR is misleading – Tiny model appears to have perfect precision, but since it doesn't make any detections at all then the FPR of it is zero, since it made no attempt at detecting anything.

A lot of the results are same for both models, being 1.0, which shows that low sample of data in some cases gives less accurate results.

3.4 Conclusions

Both whisper tiny and large models proved to be capable of detecting phrases well. Manual annotation using audiomass.co and CSV editing in VSCode was effective for creating data.

Python scripts effectively edited audio files, calculated metrics and displayed diagrams.

4. Conclusions

4.1 Design

The design worked well when used on smaller scale with pilot study to test how speech models detect phrases in different audio types. We learned that comparing tiny and large models shows differences in their performance. We learned that more audio samples are need to get more accurate results.

4.2 Pilot Study

Pilot study showed that Large Whisper model can perform better when it comes to detecting phrases than tiny model. We learned that testing approach of different audio types (clean, noisy, with different accent, emotions) works well to test model performance. Manual annotation was time-consuming, but python scripts saved a lot of time.

5. Literature

- Noisy and clean sets - https://datashare.ed.ac.uk/handle/10283/2791
- Accent set - https://www.kaggle.com/datasets/rtatman/speech-accent-archive?resource=download
- Emotion set - https://www.kaggle.com/datasets/tli725/jl-corpus
- https://arxiv.org/abs/2305.13516 - Scaling speech technology to 1000+ languages
- https://www.sciencedirect.com/science/article/pii/S1877050922014338 - Evaluation of the fficiency of state-of-the-art Speech Recognition engines
- https://ieeexplore.ieee.org/document/10348186 - A study of Audio-to-Text Conversion Software Using Whispers Model
- https://ieeexplore.ieee.org/document/10721926 - WhisperSum: Unified Audio-to-Text Summarization

- https://ieeexplore.ieee.org/document/10544133 - Speech Recognition Paradigms: A comparative Evaluation Of SpeechBrain, Whisper and Wav2Vec2 Models
- https://ieeexplore.ieee.org/document/10800465 - Evaluating Automatic Transcription Models Utilising Cloud Platforms
- https://ieeexplore-1ieee-1org-1000007600031.han.bg.pg.edu.pl/stamp/stamp.jsp?tp=&arnumber=10493971 - Automatic and Multilingual Speech Recognition and Translation by using Google Cloud API
- https://arxiv.org/abs/2006.11477 - Wav2vec 2.0: A framework for Self-Supervised Learning of Speech Representations
- https://ieeexplore.ieee.org/document/1318504 - Is word error rate a good indicator for spoken language understanding accuracy