Engkanchanith Tan
005834717

Lab 2: Communication & Speech

Task 1:

```
(180DA-WarmUp) kanchanith@tans-macbook 180DA-WarmUp % python subscriber.py
Connection returned result: 0
Received message: b'0' on topic TAN with QoS 1
Received message: b'0' on topic TAN with QoS 1
Received message: b'3' on topic TAN with QoS 1
Received message: b'6' on topic TAN with QoS 1
Received message: b'9' on topic TAN with QoS 1
Received message: b'12' on topic TAN with QoS 1
Received message: b'15' on topic TAN with QoS 1
Received message: b'18' on topic TAN with QoS 1
Received message: b'21' on topic TAN with QoS 1
Received message: b'24' on topic TAN with QoS 1
Received message: b'27' on topic TAN with QoS 1
Received message: b'30' on topic TAN with QoS 1
```

My teammate (Enkh-Uchral Enkhbayar, Austin Liu) and I tried three way MQTT communication. We all subscribe to our own topic. Austin published to my topic. I published to Eddie topic. Eddie published to Austin topic. We only publish when we receive a message in a form of number. We started from 0 and we increment by 1 everytime someone publish a new message. As a result, we were able to count the number of messages. I didn't quite notice any lagging going on.

For the early development of our project, I think we can use MQTT to communicate between three different computers. Two of the computers will behave as players and one computer will act as the server. The player computers will publish activities that are going on throughout the game while being subscribed to server's topics for updates on the scoring and progress. The server computer, on the other hand, will publish the scoreboard and subscribe to player's topics to get information for calculation.

Making MQTT automatic can be challenging. We had to figure out how to avoid running into infinite loops, in the case where a client is subscribing and publishing to the same topic. There isn't any long delay going on when we tried it out. We are planning to switch to controllers after the prototype of the game is done, which might cause more delay. I think we still need to familiarize ourselves more and see what can be done and how to best optimize it.

Task 2:

Test result:

```
Say something!
Got it! Now to recognize it...
result2:
{   'alternative': [   {   'confidence': 0.96990603,
                           'transcript': 'dog cat dog cat dog cat'},
                       {'transcript': 'dog cat dog cat doll cat'},
                       {'transcript': 'dog cat dog cat dog cat cat'},
                       {'transcript': 'dog cat dog CatDog cat'},
                       {'transcript': 'doll cat dog cat dog cat'}],
    'final': True}
You said dog cat dog cat dog cat
```

It was able to differentiate dog and cat well.

- More similar words
    - Sound and found: First try, it mistakened found for sound. Second try, it got the words right. Both had lower confidence compared to dog and cat example. However, when I put them in a sentence, all alternatives got the words right with high confidence.

```
Say something!
Got it! Now to recognize it...
result2:
{   'alternative': [   {   'confidence': 0.89092422,
                           'transcript': 'sound and sound'},
                       {'transcript': 'sound and found'},
                       {'transcript': 'sound and phone'},
                       {'transcript': 'sound & Sound'},
                       {'transcript': 'found and found'}],
    'final': True}
You said sound and sound
```

```
Say something!
Got it! Now to recognize it...
result2:
{   'alternative': [   {   'confidence': 0.9721505,
                           'transcript': 'it sounds like you found something'},
                       {'transcript': "it sounds like you've found something"}],
    'final': True}
You said it sounds like you found something
```

```
Say something!
Got it! Now to recognize it...
result2:
{   'alternative': [   {'confidence': 0.91464484, 'transcript': 'sound found'},
                       {'transcript': 'soundfound'},
                       {'transcript': 'sound phone'},
                       {'transcript': 'found'},
                       {'transcript': 'sound fun'}],
    'final': True}
You said sound found
```

    - Abcdefg works well even when I said it fast or when I sang it (although sometimes with lower confidence). However, when I say a single letter, it mostly failed to recognize A B D E F. It recognized C and G with low confidence.

```
result2:
{   'alternative': [   {   'confidence': 0.89265382,
                           'transcript': 'a b c d e
                       {'transcript': 'ABCD EFG'},
                       {'transcript': 'ABCDEFG'},
                       {'transcript': 'abcdefgh'},
                       {'transcript': 'ABCD e f g'}]
    'final': True}
You said a b c d e f g
```

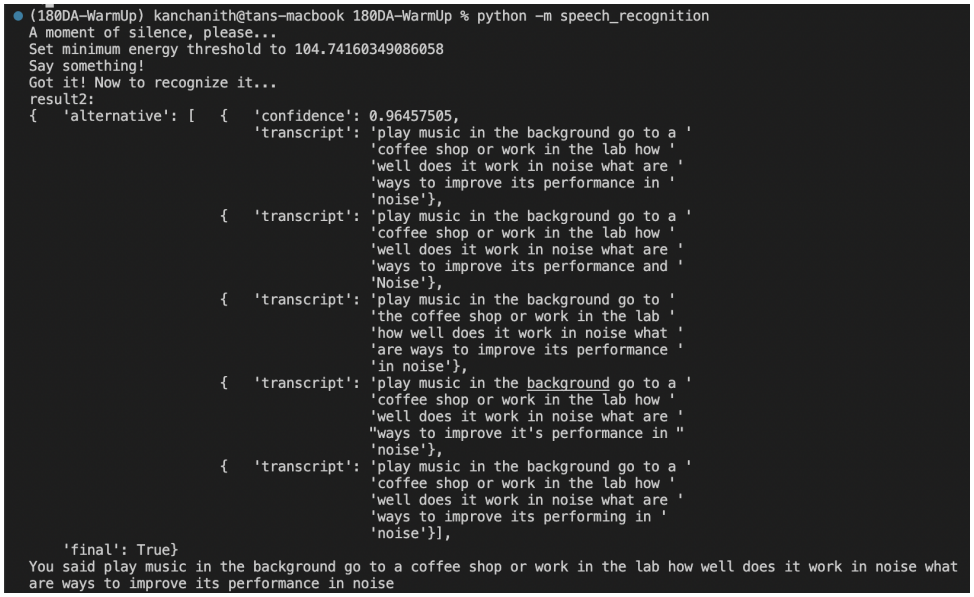
```
result2:
{   'alternative': [   {'confidence': 0.8917228,
                       {'transcript': 'gee'},
                       {'transcript': 'ghee'}],
    'final': True}
You said G
```

```
result2:
{   'alternative': [   {'confidence': 0.71362621, 'transcript': 'see'},
                       {'transcript': 'C'},
                       {'transcript': 'sea'},
                       {'transcript': 'SI'}],
    'final': True}
You said see
```

- Long Phrases: long phrases are always better. It helps for better error correction

```
result2:
{   'alternative': [   {   'confidence': 0.96841061,
                           'transcript': 'phrases how long of a phrase can '
                                         'work is the length of a phrase '
                                         'actually a good thing for error '
                                         'correction'},
                       {   'transcript': 'phrases how long of a phrase can '
                                         'work is the length of the phrase '
                                         'actually a good thing for error '
                                         'correction'},
                       {   'transcript': 'phrases how long of a phrase can '
                                         "work it's the length of a phrase "
                                         'actually a good thing for error '
                                         'correction'},
                       {   'transcript': 'phrases how long of a phrase can '
                                         'work is the length of a phrase is '
                                         'actually a good thing for error '
                                         'correction'},
                       {   'transcript': 'phrases how long of a phrase can '
                                         'work is the length of a phrase '
                                         'actually a good thing for Eric '
                                         'correction'}],
    'final': True}
You said phrases how long of a phrase can work is the length of a phrase actually
 a good thing for error correction
```

- Performed well with music in the background. I had to speak louder for it to recognize that I'm talking.



1. We're planning to use speech recognition to "switch" between weapons/tools for our game and do "power up" to boost a player's energy.
2. At first I thought one word is simple enough to avoid mistakes. After testing some words, phrases, and long sentences, I know now that long sentences have the best accuracy. A single syllable is tricky to get. A single word with few syllables works better. "Power up" had three alternatives every time I tried it. So, we can either come up with a better word or give it a pass as long as it detects the word "power."
3. Our game does not rely too heavily on the speech recognition. It takes around 1-2s to process the word "switch" and "power up." In some cases, power up can take longer to process. For example, when I tried to speak slower for better accuracy. When I speak fast, the speech program can still detect my words.
4. I don't really know the answer to this one, but my guess is that it needs a reasonable microphone and it would help to avoid too much background noise. Team players communication might also interfere with the program, especially if the sentence is long. The program might be busy processing the sentence and not able to catch the keywords. To avoid this problem, the mic has to be mounted at a stationary place and not on the players. We can also condition the speech processing time to a certain value to avoid processing longer sentences.

Task 3:

www.github.com/Kanchanith/180DA-WarmUp/