

Voice Recognition



Please replace the OpenAI API key in the code with your own.

Feature Name: Voice Recognition

Version	1.0.0
Date	29 Oct 2023
Author(s)	Sicheng NIE

Overview

app, s_to_t, t_analysis nodes under **my_pac** package attempts to translate the voice to meaningful command. The final command information would be published to a topic named **translated_sentence**.

Prerequisites

- Python 3.x installed
- ROS2 environment
- Useable micro
- Install the following packages via pip

```
import rclpy
import std_msgs.msg
import tkinter
import sounddevice
import wavio
import openai
```

Usage

1. Setting up the Workspace

1. Navigate to the ROS2 workspace

```
cd cd ros2_workspace
```

2. Build packages

```
colcon build --packages-select my_pac
```

3. Source the setup file

```
source install/setup.bash
```

2. Run an automation script. This will run all required nodes.

```
ros2 launch my_pac nodes_launch.py
```

3. Check Result

The results of each node and the final generated results will be displayed on the terminal.

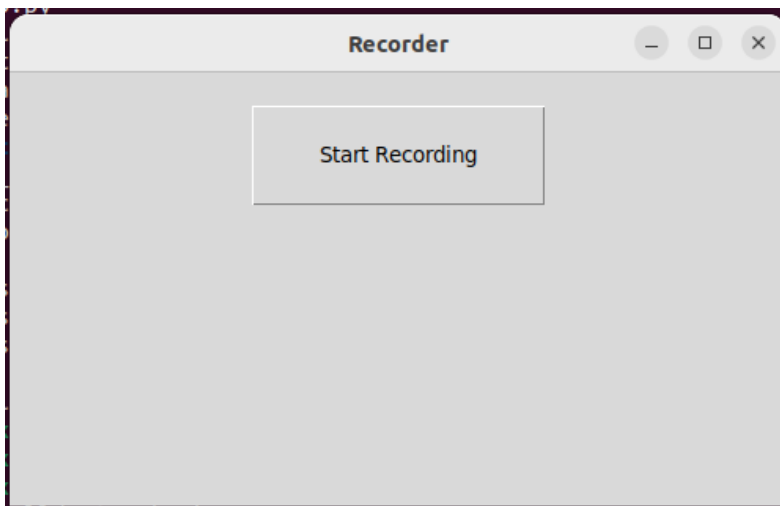
```
[s_to_t-2] [INFO] [1697089234.154852177] [s_to_t]: Original Transcript: Move that pen to the top of the table.
[s_to_t-2]
[s_to_t-2] [INFO] [1697089234.629835045] [s_to_t]: Corrected Transcript: Move the pen to the top of the table.
[t_analysis-3] [INFO] [1697089236.032166871] [t_analysis]: Action: Move
[t_analysis-3] [INFO] [1697089236.032567226] [t_analysis]: Object: pen
[t_analysis-3] [INFO] [1697089236.033115366] [t_analysis]: Location: top of the table
[t_analysis-3] [INFO] [1697089236.033466919] [t_analysis]: Cost would be 79
```

You can also listen to the final result on another terminal by using the following command.

```
ros2 topic echo /translated_sentence
```

Process

1. The **app** node will open a GUI and wait for the user's voice input. After the user completes a voice input, publish the address of the voice on **audio_filepath** topic. The program only maintains one voice file.



2. The **s_to_t** node will access the voice file of the monitored address, convert the voice into text through OpenAI, and make corrections to the text. Finally, publish the text on the **input_sentence** topic.

3. The **t_analysis** node will perform semantic analysis on the text, extract corresponding keywords and information, and ultimately publish them on the **translated_sentence** topic.

Release Notes

Features:

- Receive the voice input from the user through the microphone, first convert the voice into text, and then perform semantic analysis on the text in a fixed format
- Information is transmitted between multiple ROS2 nodes
- Only the latest voice will be captured

Known Issues:

Holding down the spacebar for continuous recording is not supported due to keyboard auto-repeat function challenges.

Please build the workspace and ROS2 package according to the document. The package structure is as follows:

Notes:

- When recording, please reserve 0.5 seconds after the end of the speech to ensure that the audio can be fully saved.
- Please follow the normal format for the input voice.
- Since the system currently only maintains one voice file, please do not attempt to input multiple voices simultaneously.

Future Improvements:

1. Add the function of viewing history. Increase the number of voice files maintained. Users can view the previous voice and analysis results.
2. Add a wider range of paradigms to allow users to input more freely.