MaAI

A **Real-time** and **Light-weight** Software for Generation of **Non-Linguistic** Behavior Generations for Conversational AIs

(Real-time Implementation of Voice Activity Projection)

📄 README: English | Japanese (日本語)

MaAI is a state-of-the-art and light-weight software that can generate (predict) non-linguistic behaviors in real time and continuously. It supports essential interaction elements such as **turn-taking**, **backchanneling**, and **nodding**. Currently available for English GB, Chinese CN, and Japanese JP languages, MaAI will continue to expand its language coverage and non-linguistic behavior repertoire in the future. Designed specifically for conversational AI, including spoken dialogue systems and interactive robots, MaAI handles audio input effectively in either two-channels (user-system) or single-channel (user-only) settings. 🎙 Thanks to its lightweight design, MaAI operates efficiently, even exclusively on CPU hardware. ⚡

The currently supported models are mainly based on the Voice Activity Projection (VAP) model and its extensions. Details about the VAP model can be found in the following repository: 🔗 https://github.com/ErikEkstedt/VoiceActivityProjection

🆕 Update

- 🚀 We launched the MaAI project and repository here! (July 14th, 2024)

🚀 Getting Started

To quickly get started with MaAI, you can install it using pip:

```
pip install maai
```

> 💡 **Note:** By default, the CPU version of PyTorch will be installed. If you wish to run MaAI on a GPU, please install the GPU version of PyTorch that matches your CUDA environment before proceeding.

You can run it as follows. 🏃 The appropriate model for the task (mode) and parameters will be downloaded automatically.
Below is an example where two wav files (user and system) are input to the turn-taking model (VAP).

```
from maai import Maai, MaaiInput

wav1 = MaaiInput.Wav(wav_file_path="path_to_your_user_wav_file")
wav2 = MaaiInput.Wav(wav_file_path="path_to_your_system_wav_file")

maai = Maai(mode="vap", frame_rate=10, context_len_sec=5, audio_ch1=wav1,
audio_ch2=wav2, device="cpu")

maai.start_process()
while True:
    result = maai.get_result()
```

🧩 Models

We support the following models (behavior, language, audio setting, etc.), and more models will be added in the future. 🆙

Turn-Taking

The turn-taking model uses the original VAP as is and predicts which participant will speak in the next moment.

- VAP Model
  - Japanese
  - English
  - Chinese
- Noise-Robusst VAP Model (**Recommended**)
  - Japanese
- Single-Channel VAP Model
  - In Preparation ...

Backchannel

Backchannels are short listener responses such as `yeah` and `oh`, that are also related to turn-taking.

- VAP-based Backchannel Prediction Model
  - Japanese - Timing Only
  - Japanese - Timing for Two types
- Noise-Robusst VAP-BC
  - In Preparation ...
- Single-Channel VAP-BC
  - In Preparation ...

Nodding

Nodding refers to the up-and-down movement of the head and is closely related to backchanneling. Unlike backchannels that involve vocal responses, nodding allows the listener to express their reaction non-verbally.

- VAP-based Nodding Prediction Model
  - Japanese

🛗 Input / Output

For input to the MaAI model, you can directly call the `process` method of a `Maai` class instance.
The `MaaiInput` class also provides flexible input options, supporting audio from WAV files, microphone input, and TCP communication.

- WAV file input: `Wav` class 📁
- Microphone input: `Mic` class 🎙️
- TCP communication: `TCPReceiver` / `TCPTransmitter` classes 🌐

By using these classes, you can easily adapt the audio input method to your specific use case.

For output, the `MaaiOutput` class is currently under development.
At present, you can retrieve the processing results using the `get_result` method of the `Maai` class instance.

For more details, please refer to the following README files:

- Input Readme
- Output Readme

💡 Example Implementation

You can find example implementations of MaAI models in the test_scripts directory of this repository.

- Turn-Taking (VAP)
  - With 2 wav file inputs 🎧
  - With 2 mic inputs 🎤
  - With 2 mic inputs via TCP networks 🌐
  - With 1 wav file adn 1 mic inputs 🎧🎤
- Backchannel
  - With 1 wav file adn 1 mic inputs 🎧🎤
- Nodding
  - With 1 wav file adn 1 mic inputs 🎧🎤

📑 Publication

Please cite the following paper, if you made any publications made with this repository. 🙏

Koji Inoue, Bing'er Jiang, Erik Ekstedt, Tatsuya Kawahara, Gabriel Skantze
Real-time and Continuous Turn-taking Prediction Using Voice Activity Projection
International Workshop on Spoken Dialogue Systems Technology (IWSDS), 2024
https://arxiv.org/abs/2401.04868

```
@inproceedings{inoue2024iwsds,
    author = {Koji Inoue and Bing'er Jiang and Erik Ekstedt and Tatsuya Kawahara and Gabriel Skantze},
    title = {Real-time and Continuous Turn-taking Prediction Using Voice Activity Projection},
    booktitle = {International Workshop on Spoken Dialogue Systems Technology (IWSDS)},
    year = {2024},
    url = {https://arxiv.org/abs/2401.04868},
}
```

If you use the multi-lingual VAP model, please also cite the following paper.

Koji Inoue, Bing'er Jiang, Erik Ekstedt, Tatsuya Kawahara, Gabriel Skantze
Multilingual Turn-taking Prediction Using Voice Activity Projection
Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING), pages 11873-11883, 2024
https://aclanthology.org/2024.lrec-main.1036/

```
@inproceedings{inoue2024lreccoling,
    author = {Koji Inoue and Bing'er Jiang and Erik Ekstedt and Tatsuya Kawahara and Gabriel Skantze},
    title = {Multilingual Turn-taking Prediction Using Voice Activity Projection},
    booktitle = {Proceedings of the Joint International Conference on
Computational Linguistics and Language Resources and Evaluation (LREC-COLING)},
    pages = {11873--11883},
    year = {2024},
    url = {https://aclanthology.org/2024.lrec-main.1036/},
}
```

If you also use the noise-robusst VAP model, please also cite the following paper.

Koji Inoue, Yuki Okafuji, Jun Baba, Yoshiki Ohira, Katsuya Hyodo, Tatsuya Kawahara
A Noise-Robust Turn-Taking System for Real-World Dialogue Robots: A Field Experiment
https://www.arxiv.org/abs/2503.06241

```
@misc{inoue2025noisevap,
    author = {Koji Inoue and Yuki Okafuji and Jun Baba and Yoshiki Ohira and
Katsuya Hyodo and Tatsuya Kawahara},
    title = {A Noise-Robust Turn-Taking System for Real-World Dialogue Robots: A
Field Experiment},
    year = {2025},
    note = {arXiv:2503.06241},
    url = {https://www.arxiv.org/abs/2503.06241},
}
```

If you also use the backchannel VAP model, please also cite the following paper.

Koji Inoue, Divesh Lala, Gabriel Skantze, Tatsuya Kawaharaa
Yeah, Un, Oh: Continuous and Real-time Backchannel Prediction with Fine-tuning of Voice Activity Projection
https://aclanthology.org/2025.naacl-long.367/

```
@inproceedings{inoue2025vapbc,
    author = {Koji Inoue and Divesh Lala and Gabriel Skantze and Tatsuya
Kawahara},
    title = {Yeah, Un, Oh: Continuous and Real-time Backchannel Prediction with
Fine-tuning of Voice Activity Projection},
    booktitle = {Proceedings of the Conference of the Nations of the Americas
Chapter of the Association for Computational Linguistics: Human Language
Technologies (NAACL)},
    pages = {7171--7181},
    year = {2025},
    url = {https://aclanthology.org/2025.naacl-long.367/},
}
```

📝 License

The source code in this repository is licensed under the MIT license. The trained models, found in the asset directory, are used for only academic purposes.

A pre-trained CPC model, located at `asset/cpc/60k_epoch4-d0f474de.pt`, is from the original CPC project and please follow its specific license. Refer to the original repository at https://github.com/facebookresearch/CPC_audio for more details.