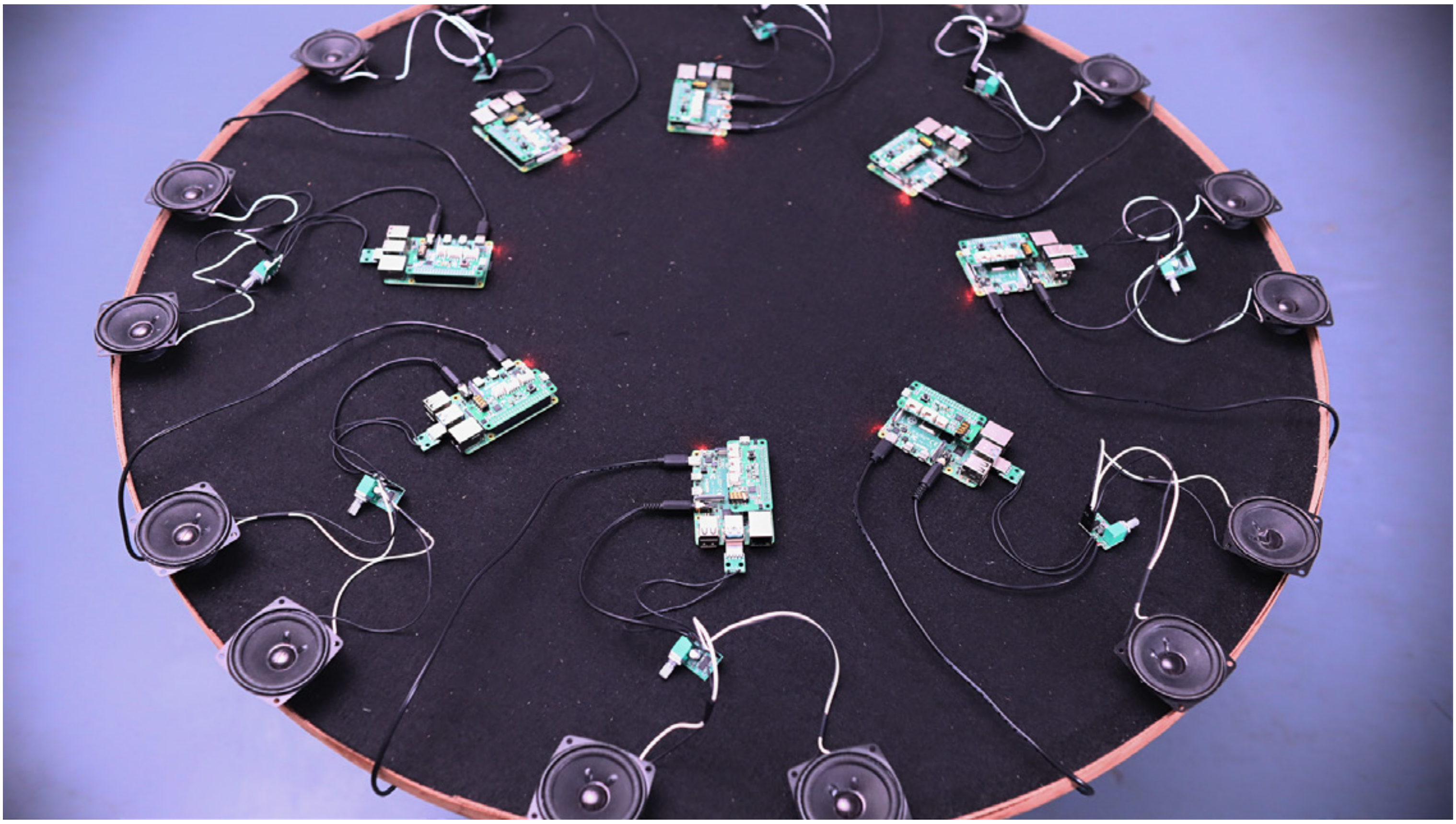
xCoAx 2022 10th Conference on
Computation, Communication, Aesthetics & X
2022.xCoAx.org
Coimbra, Portugal



Ich Sage Was Ich Sagen Kann (ISWISK): Voice Composition by Real-time Automatic Speech Recognition

Hyungjoong Kim
hjkaddict@gmail.com
Berlin University of the Arts,
Berlin, Germany

Ich Sage Was Ich Sagen Kann is a voice composition experiment, where multiple self-made voice bots have a dialogue with each other by using Keyword Spotting (KWS) in the Automatic Speech Recognition (ASR) domain. The dialogue is not considered as a conversation between different personalities, but it reflects on a process of uttering thoughts from my personal experience. I make speech recognition modules by using a Convolutional Neural Network (CNN) model installed on eight microcontrollers, which give utterances when certain keywords are detected. The composition consists of four parts, from polite one-on-one conversation between myself and one bot to the impolite conversation between 8 bots, which reflects the chaotic state of mind in the thinking process. This practice aims not only at overcoming problems in communication and learning of a foreign language at a personal level, but also at the possibility of communication with human language between machines.

Keywords Uttering Mind, Voice Composition, Speech Recognition, Keyword Spotting

DOI 10.24840/xCoAx_2022_65

Background and Idea

Ich Sage Was Ich Sagen Kann is a voice composition experiment, where multiple self-made voice bots have a dialogue with each other by using Keyword Spotting (KWS) in the Automatic Speech Recognition (ASR) domain. The dialogue is not considered as a conversation between different personalities, but it reflects on a process of uttering thoughts from my personal experience. I personally have difficulties in communicating with others in speech. The reason that I came up with, in my mind, was those secondary thoughts, which are derived from the initial thought, could act as a resistance to uttering what is on my mind. Let's consider the situation where my colleague and I are having a conversation. After he asks "What do you think about this sound?", I could possibly reply "I think this sound is really interesting". But even before the thought is uttered, secondary thoughts such as "What does this mean, "interesting"?", "What is sound?", "How do you define interesting sound?" are spun off from the first idea, then the thoughts and speech end up being jumbled and disconnected.

I would call this a form of negative feedback in the process of uttering the mind. Negative feedback occurs when some function of the output is fed back into the input to reduce the fluctuations in the output, and it usually happens in organic systems for balancing their processes. When it comes to uttering the mind, however, I experienced that this feedback process causes a severe self-censorship problem and gives me a hard time when communicating with others. Therefore, acknowledging (or possibly overcoming) this difficulty was the initial motivation of starting this artistic practice as a kind of self-therapy method.

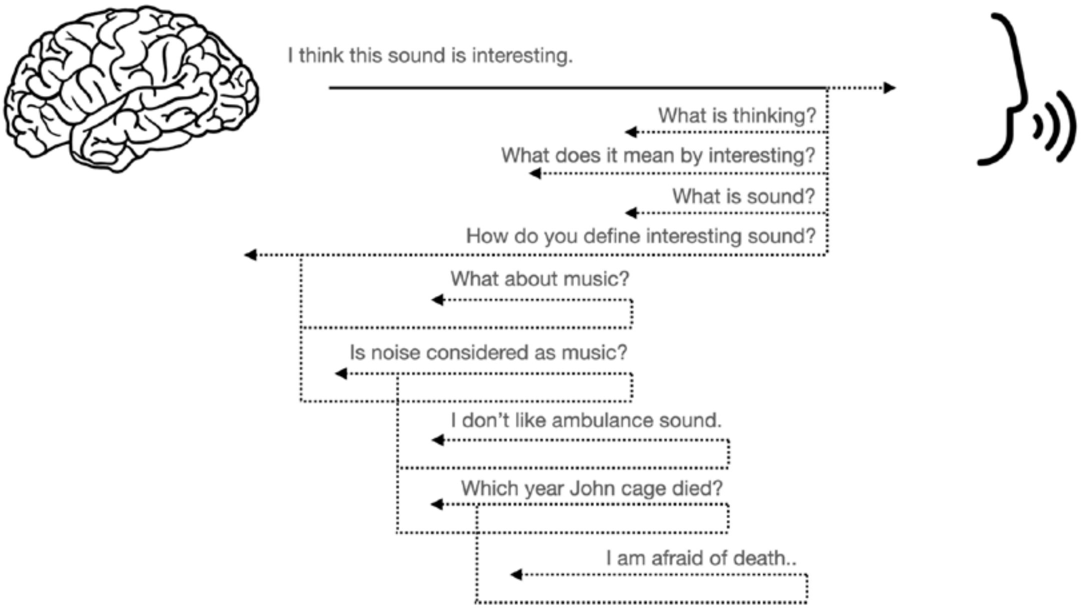

I think this sound is interesting.
What is thinking?
What does it mean by interesting?
What is sound?
How do you define interesting sound?
What about music?
Is noise considered as music?
I don't like ambulance sound.
Which year John cage died?
I am afraid of death..

Fig. 1. The uttering mind problem.

Keywords and Utterances

In this practice, I would like to speak out every thought, including initial and secondary ones, rather than let those fade out internally, which means turning negative feedback into positive expression. Most importantly, the realization of this thinking process as a voice composition is the main purpose of this work.

The implementation plan for achieving this is using multiple voice bots which have speech recognition functions. If one thought is uttered initially from one bot, secondary utterances are followed from the other bots based on recognizing speech in the first utterance. This process could be interesting not only for algorithmic composition itself, but also for experimenting with communication processes between voice bots. The speech recognition modules that I built in this project are based on Keyword Spotting (KWS), which deals with the classification of keyword in utterances. The reason for choosing this method is that I tend to obsess about the lexicon more than the actual context of the full sentence.

Now, which keywords should be recognized, and which sentences should be uttered? The starting point was writing down simple utterances that came out from the thinking process. I start with 'Ich sage was ich sagen kann.' (I say what I can say). From this sentence, I choose an arbitrary word 'sagen' and write down a few sentences that come to me when I think of that word. For example,

× *Mein Gedanken auszusprechen ist nicht einfach.*
(Speaking out my thought is not easy.)

× *Als ich jung war, sollten wir in der Öffentlichkeit nicht meine eigene Gedanken äußern.*
(When I was young, we were not supposed to express our own thoughts in public.)

× *Manchmal habe ich Angst wenn ich etwas sagen muss.*
(Sometimes I am afraid when I have to say something.)

× *Wenn ich spreche, überlege ich mich ob das richtig ist.*
(When I speak, I consider whether it is right.)

Imagine a situation when one bot speaks a word or a sentence including 'sagen'. If the speech recognition module is properly installed, the other bots can recognize the keyword and answer with one of those sentences. But then, what happens if the other bot says, "Mein Gedanken auszusprechen ist nicht einfach."? (First sentence above). Nothing will be triggered because bots cannot recognize any words in that given utterance. For achieving ping-pong like conversation, I continue to choose words from the sentences, collect them as recognizable keywords, and write down corresponding sentences. Throughout this process, I collect 30 keywords and write 103 utterances (3 to 4 sentences on each keyword): 'Gedanken', 'äußern', 'Spaß', 'Freiheit', 'einfach', 'sympathisieren', 'nachdenken', 'aussprechen', 'geniessen', 'rauchen', 'schwierig', 'Gesundheit', 'Öffentlichkeit', 'Rückkopplung', 'Verlangen', 'sagen', 'ausdrucksvoll', 'essen', 'richtig', 'klar', 'Fehlern', 'Angst', 'umgeben', 'wissen', 'KI', 'glücklich', 'nervös', 'überlegen', 'kacken', 'schreiben']

Technical Realization

To make a simple keyword spotting module I proceeded in four steps: 1) Recording the set of keywords in my voice, 2) Pre-processing the dataset, 3) Training the Neural Network Model, and 4) Compiling the model and running it on a microcontroller for real-time recognition.

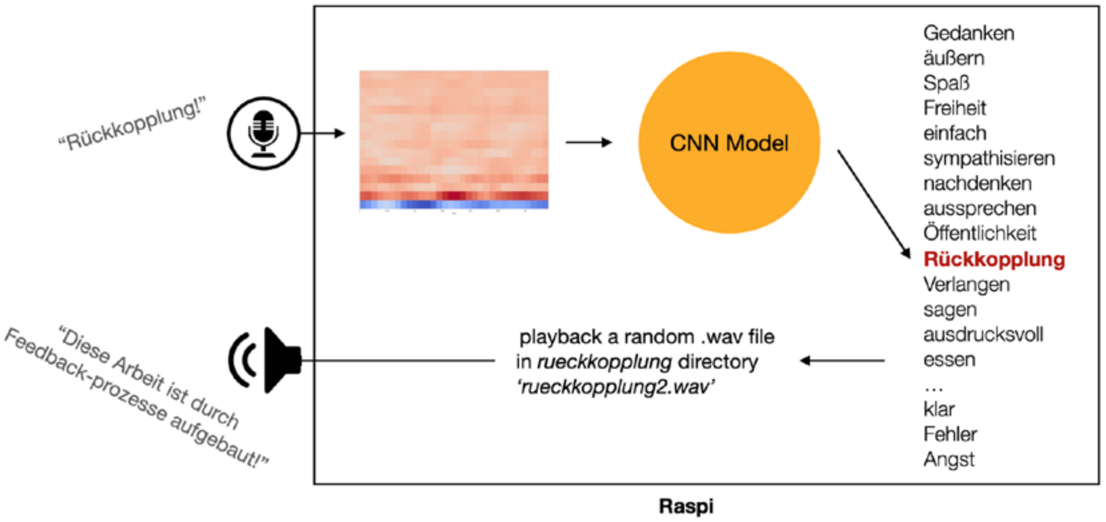


Fig. 2. The schema of real-time keyword spotting module.

In the first step, I recorded each keyword 300 times by using my voice. 100 times with the *MacBook microphone*, another 100 times with *ReSpeaker 2-mic HAT* which I use in the actual recognition module. For the rest, I played back the recorded files with a speaker and recorded them back into the *ReSpeaker 2-mic HAT*, so that modules could learn to recognize the voice coming out from the speaker. The sample rate is always set to 16kHz. Background noise is also added as a keyword in dataset for not giving a confusion in classification.

Secondly, pre-processing the dataset is needed. Since all recordings have different durations, I unify the length of the samples to 1 second. This step is necessary because the data shape of the neural network has to be identical. This is done simply by padding zero to rest of the array for shorter than 1-sec files and cutting the rest for longer than 1-sec files. After unifying the lengths, I extracted MFCCs (Mel Frequency Cepstral Coefficients) features from the dataset, which aims at developing the features from the audio signal that can be used for detecting the phonemes in speech.

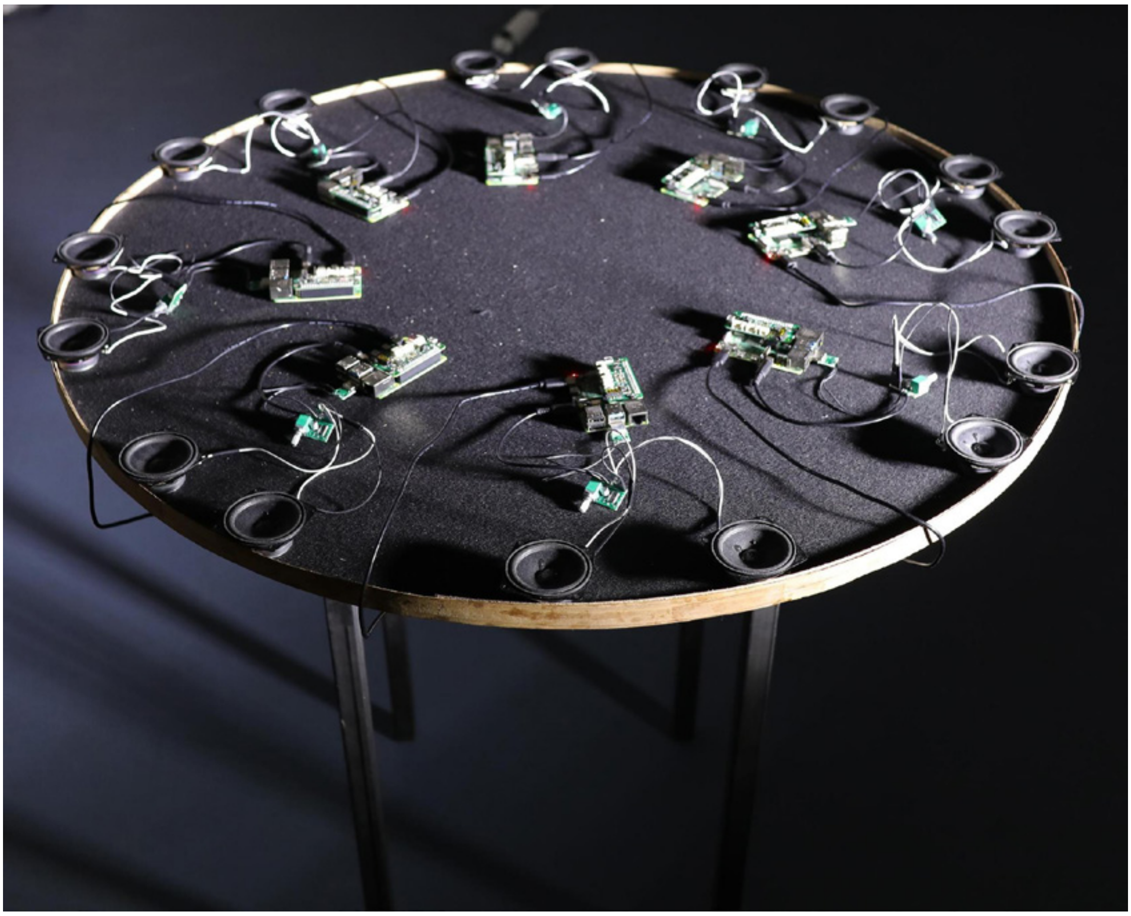
Thirdly, splitting the dataset into training (80%), validation (10%), and test (10%) is what follows for the training step. I use a 3-layered convolutional neural network model. The validation accuracy, meaning the prediction rate of classification, is 96 percent, which is quite high, but expectable at the same time because the dataset only consists of my own voice.

After training the model, I compiled the model in .tflite format and ran it on *Raspi 4B*. In this step, I used the *sounddevice* library for dealing with the input stream. The input signal is consistently stored in an array and the microcontroller runs a classification every 0.5 seconds. If it successfully classifies a certain keyword, it plays the corresponding utterance, which I recorded in advance.

Composition

The voice compositions I have been experimenting with in this project mainly consist of two modes: One is polite conversation and the other is impolite conversation. Technically, this is distinguished whether the module has a function of voice activity detection. In polite conversation, bots mute themselves until the other speech they hear is over. In impolite mode, on the other hand, this function is absent, so the bots will begin to utter right after recognizing a keyword.

Fig. 3. Installation View.



At the beginning of the composition I participate live in the conversation. My role is triggering the bots by saying a certain sentence or keyword when no bot recognizes any keywords from the other bots. This non-recognition situation happens quite often, and I assume that there are two critical reasons. First, the dataset is not strong enough to deal with unexpected background noise and room reverberation. The dataset I used in this project is certainly limited in terms of diversity of the data since it only consists of my voice, which I recorded in the same room. Secondly, there must be a difference between my raw voice and the playback voice coming out from speakers. Supposedly, the machine can also distinguish between raw voice and speaker voice, as we all do. In this experiment, unfortunately I could not deal with these limitations. This is only a problem if we consider this experiment as being about full functionality, but I think this malfunction is actually an interesting and fun part of making artworks from machine learning.

So far, I have created 4 compositions by setting up different topologies and behaviours:

1. In the 1 bot and myself composition, the bot and I are having one-on-one chat as a prelude.

2. In the 4 bots and 1 conductor bot composition, a conductor bot takes the leading role I performed in composition 1. When a silence occurs, the conductor bot plays a randomly chosen keyword to continue the conversation.

3. In the 7 bots and 1 conductor bot composition, 3 more bots come into play. This is considered as an impolite conversation because they speak out right away once it recognizes a keyword; this reflects more closely on the chaotic state of mind in the thinking process described.

4. In the 8 bots composition, each bot has a role of conductor and participant at the same time. When the bot does not detect any keyword, a randomly chosen keyword is played.