Tweet Sonifier

an interactive media piece by Sam Smith,
Sophie Mirza, Joey Fortino, and Geoff Brown

introduction

This installation was designed for the final project of an introductory computer programming course. The project prompt was to create a program that utilized some form of artificial intelligence or machine learning. We chose to create a product that sonified tweets sent to our twitter account, and then broadcasted the sound to the user via headphones. The program's audio response was created by resampling existing audio files based on natural language processing information from the tweet. This project utilized python, Twitter's API, pyaudio, and portaudio. The project was inspired by musical pieces with alternative notation such as pictures and or drawings; our project utilized tweets as an alternative medium.

Our project relies on two primary pieces of media to operate-- a source text and audio files. In the original presentation, we chose to use Moby Dick by Herman Melville as the source text and a variety of isolated audio files (drums, keyboard, vocals, piano, etc.) as the source audio. We chose the classic novel for its antiquated language and unique subject matter. Additionally, this text is lengthy and reflects common language phenomenons such as high frequency of words such as "and" and "the." The audio files were taken from a variety of sources such as commercial recordings and personal recording projects, all of which had a unique timbre and feel.

the result

The final sonic result, outwardly having little to do with our intertextual deconstructions, actually does the same thing with music. It takes the natural, spontaneous inclinations of improvisation in music, hearkening back to the texts of madrigals, and assigns them an objectivity, in frequency base and amplitude, set by the computer. The intertextuality between incoming Tweets and Moby Dick also exists as a dichotomy between improvisational spontaneity and rigid counterpoint or aleatoric composition: this schism is represented by the loud, noisy content of our source files - while each file maintains its own internal consistency, the compilation of disparate parts sounds cacophonous. They depict that the need to make, that drives creativity, is not objective.

process

The first step in our code was to destroy Moby Dick. The code mangles the writing - the ordering of words, spacings, capitalization, punctuation and all context of the words are stripped This reduces the intent of the novel to the means by which the intent is portrayed: words themselves. Each word is stripped of their contextual meaning and is exchanged objectivity, in the form of a number. This reduction allows for a formulaic base by which to evaluate through our lens of Tweets. The suggestion is that any text could have been used, as each word in it is just that - a word. While other books probably won't use "whale" or "harpoon" or "sea" as much, the words which are most likely to be tweeted from our account are the most generic - "the," "and," etc. Evaluating a work for its words and not for its meaning yields nothing. Context gives meaning, and that meaning is unstable and multiple, as is one of the core arguments for Post-Structuralist philosophy.

Upon initialization, the program loads files in audioFiles folder. It takes the guitar, bass, drum, and vocal files and splits them into 10 second chunks, performing a Fast-Fourier Transform (FFT) on the chunks and assigning each chunk two values between 0 and 1 , which correspond to Frequency and Amplitude. It cleans the .txt file ("Moby Dick.txt") to strings with no special characters and trains the Language Model on the text. Based on the incoming Tweet stream, it assigns each Tweet a value based on the average frequency of the words in the tweet by comparing them to the language model and scales. This is then compared to the Frequency for guitar and bass files, and amplitude for drum and vocal files. These are layered on top of each other and then played back.

more info

project prompt - https://eecs183.org/showcase