*Twitthear*: An Audio Interface for Twitter

| Ryan Rose  Georgia Institute of Technology  Atlanta, USA  rrose37@gatech.edu |
| --- |

# ABSTRACT

Twitthear is an audio interface that brings Twitter to the virtual personal assistant Amazon Alexa. Its goal is to encode information about a tweet in musical phrase for a user to listen to, and decide whether or not to save the tweet to read on a visual interface later. Using sentiment analysis, semantic rhythm, and various sonification techniques to create meaningful melodies, Twitthear seeks to be more useful than traditional text-to-speech social media readers.

# Concept

Today’s tech world has been noticeably impacted by the dawn of the virtual personal assistants such as Amazon Alexa, Apple Siri, Google Assisstant, and Microsoft Cortana. These devices operate almost entirely in the audio domain, with users speaking commands and receiving responses in computer-synthesized human speech. The rise of audio interfaces presents a new platform for many applications which have previously been confined to point-and-click input and screen-based output. This is the case with social media, which currently relies almost entirely on visual domain content such as text, images, and videos. The goal of this project, Twitthear, is to develop an audio domain application for Twitter [9], suitable for use with virtual personal assisstants.

While a quick port of Twitter to the audio domain might just be a text-to-speech (TTS) representation of tweets, Twitthear seeks to provide a more useful Twitter interface through music. One of issue with TTS is the time it takes to read an entire tweet. Furthermore, in order for any sentiment value to be taken from listening to a TTS converted tweet, the entire tweet must be listened to with full attention. Twitthear first addresses these concerns by providing musical representations that are shorter than the time it takes to read a tweet aloud. Additionally, music capitalizes on the incredible ability of the human brain to detect patterns. Musical patterns are leveraged to provide quick access to identifying information about a tweet: for example, a collection of three dissonant eighth notes can express negative sentiment in a tweet much faster than the TTS translation of a sentence, which requires every word to be read for the negative sentiment to be understood.

The previous example also brings light to the intended usage of Twitthear. The intention is not to have users move from accessing Twitter solely in the visual domain to solely in the audio domain. Rather, Twitthear uses the audio domain to curate a collection of tweets that are particularly interesting for a user, for them to read on a visual interface at a later time. This prevents the loss of visually-dependent content such as images and video.

An anticpated user of Twitthear is anyone who uses virtual personal assisstants and Twitter. The ability to prune a timeline of tweets, often filled with ads or other uninteresting content, should attract a large amount of users. An additional market for Twitthear could be the visually impaired, due to the advantages of using musical representation of tweets instead of TTS.

# Context

Twitter has been the source of input for multiple musical systems. One example is CCRMA’s *TweetDreams*: a musical performance driven by real-time collection of tweets [8]. Tweets create musical content through melodic representations, though the melodies are not reflective of the content of the tweet (such as sentiment analysis or Twitter metadata). Instead, tweets with similar content are grouped together with similar generative melodies, creating a theme-and-variation piece that can be guided by audience members publishing tweets with appropriate hashtags and content.

Twitthear also takes inspiration from The Listening Machine, which does seek to create musical reflective of emotional sentiment in a surveyed collection of tweets [6]. This system focuses on creating art based on many tweets over time, considering the rate of incoming tweets as well as keywords in its sonification processes.

Sonification for informational purposes finds consistent usage in the network security industry. Statistics like the amount of activity in a network are encoded in music through hums or tones, whose absence are more noticeable than their presence, allowing administrators to detect when networks go down [2]. Additionally, data such as IP addresses of ports connections are sonified with the intent of being distinguishable. Ballora et al. used timbral manipulation of a synthesized gong sound to represent four numbers that make up an IP address [2]. Twitthear extends this idea to sonifying tweet metadata such as retweet and favorite count through timbral-shifting effects.

Lastly, Twitthear was influenced by a colleague in the music technology field, Jack Armitage, to consider “Mickey-Mousing” musification, where syllabic patterns of text translate to musical rhythm [1].

# Implementation

## System Architecture

Twitthear is interacted with primarly through an Amazon Alexa skill, in which the user can “ask Twitthear” various commands. The primary commands are to “play my tweets,” “play tweets about” a search term, or “play tweets from” a location. Upon hearing a command, Alexa triggers a response in Twitthear’s Python script. This script houses the Twitter API interactions [5], Google Cloud Natural Language API for sentiment analysis [4], NLTK for syllable extraction [7], and other libraries for extracting tweet features for creating musical representations. After the set of tweets is collected and features have been extracted, a tweet’s data is sent to a Max/MSP patch over OSC for musification. The patch creates a MIDI representation of a musified tweet, which is sent to Ableton Live for synthesizing.

While playing, commands to Alexa to “pause,” “resume,” “skip,” and “go back” all route commands through Python to Max to control playback. The ability to have the playback controls of “skip” and “go back” led to a need for either a discrete musification mapping so that identical tweet inputs would have identical musical output, or a method to save off and recall musical phrases. The latter was implemented, using a MySQL database that is communicated with via Python. When a new tweet is sonified, its Twitter-labeled ID serves as the filename of a saved representation in Max. If that tweet ID is served up to be sonified again, the file is read from disk, and the previously made musical phrase is played again. This allowed for the inclusion of randomness in Twitthear’s musification processes.

Lastly, when a user asks to “save this tweet,” a the tweet is sent to the Twitthear visual interface over OSC. This is a simple web page served via Node.js that displays the embedded tweet. The only interaction on this interface is the ability to clear a tweet when finished reading (Figure 1).



Figure . The Twitthear visual interface.

## Musification

There are two musification processes involved in Twitthear: a discrete and repeatable musification of Twitter usernames, and a musification that incorporates randomness for tweet content. The former uses a discrete process so that every username will have the same musification. This is to allow users of Twitthear to build memory mappings of certain musical phrases to a given user. Over time, this will build the ability to recognize playing tweets published by people a user cares about. The rhythmic content of a username musification is fixed: three chords—two sixteenth note chords followed by a quarter note chord. Three chords allows for a wide array of harmonic possibilities, and the extended length of the final chord is to give users a moment to process the musification sequence they just heard. The chords’ harmonic values are dependent upon the content of the username. The process is as follows: the username first is split into two halves, each defining one chord. The first lowercase letter defines a scale from which the chord is drawn, as well as the root. Every following lowercase letter adds a third to the chord. Uppercase letters raise the octave of the chord, and underscores lower the octave of the chord. This defines two chords of the progression: the third is generated using the Cage Project’s [chordinterp] object in Max [3]. The third chord, which is played between the two previously generated, is an interpolation of the two chords using this object. Any numbers in the username contribute to the amount of interpolation.

For the tweet content musification, the process procedes thusly: first, for harmonic content, sentiment analysis is used to define interpolation between a set of five chords. The first of these chords is a major or minor triad, depending if the sentiment is positive or negative, respectively. The other four chords are stepwise variants of these chords, loosely modeling chords of scale degrees vi, vii, IV, and V. Each of these four chords are mapped to part of a sentiment quadrant: V and IV are mapped to positive sentiment, and vi and vii are mapped to negative sentiment. Within positive or negative sentiment, a chord is chosen based on the magnitude of the sentiment. With one quadrant chord chosen, notes are generated through interpolation between the major or minor triad first established, and the described chosen chord. The velocity of these notes are relative to the magnitude of the sentiment, with a small amount of random variability.

The rhythm of the tweet content is formed using the “Mickey-Mouse” method suggested by Armitage [1]. Each word of tweet text is broken into syllables, with every syllable of a word given an eighth- or sixteenth-note. Every word is also delimited by an eight- or sixteenth-note rest. This gives the effect of an accelerated rhythmic pattern of speaking the text.

Two additional pieces of tweet metadata are sonified, the number of retweets and the number of favorites the tweet has. Retweets are mapped to a delay affect in Ableton, with a higher number of retweets causing greater delay. This is to emulate the phrase being repeated, representing the tweet being shared through retweets. Favorites are expressed through a chorus effect, signifying multiple people liking the tweet and reciting it in tandem. These effects were also chosen to be uniquely distinguishable.

The arrangement of the sonified content is such: first, the username of the person who published the tweet is played, followed by the sonified content of the tweet. Lastly, usernames mentioned in the tweet content are sonified and played back. To delimit tweets, a carriage return sound is played after a tweet sonification has finished.

# Conclusions and Future Work

Twitthear excels as a very usable interface: there are a limited amount of conventional, understandable commands. The Alexa skill follows conventions of other third party skills of “ask {skill} to …,” which should facilitate ease of use. Furthermore, the visual interface is intentionally minimal, and is only used for browsing saved tweets.

The system’s output succeeds in being a succinct representation of tweets. However, the output of this system as a whole does present some concerns, and cannot yet substantiate claims of effectiveness. The primary concern is that the musical encodings of tweet data are not effectively translated to users. To test this, user study is needed. The mappings of this system are thoughtfully designed to facilitate the human memory mapping process, but this cannot be confirmed without further evaluation. One tool that could help this is the ability to play just the musification of an input username. That way, a user could learn what to listen for to distinguish tweets from people they actively follow.

On one hand, there is motivation to use the system repeatedly to understand the sonification better and get more out of its mappings. On the other hand, there is a limited range of musical output in this system that may make repeat use musically boring. This is in part because of the very purposeful mappings used to encode tweet information. Some possible solutions to this would be to either roll out different, similarly considered mapping schemes, or let users define their own mappings. There is concern that allowing users freedom to do this will lead them to inadvertently lose the sonification encodings. In all, Twitthear provides a novel interface that, while using music more as a tool than as an art, provides potential for continued use in bringing social media to the audio domain.

# REFERENCES

[1] Armitage, J. 2017. Informal feedback on Twitthear.

[2] Ballora, M. et al. 2011. Songs of cyberspace: an update on sonifications of network traffic to support situational awareness. *Proc. SPIE* (2011), 80640P–80640P.

[3] cage: 2016. *http://www.bachproject.net/cage/*. Accessed 2017-12-14.

[4] Cloud Natural Language: 2017. *https://cloud.google.com/natural-language/*. Accessed: 2017-12-14.

[5] Docs — Twitter Developers: 2017. *https://developer.twitter.com/en/docs*. Accessed: 2017-12-14.

[6] FAQs | The Listening Machine: 2012. *http://www.thelisteningmachine.org/faq*. Accessed: 2017-11-10.

[7] Natural Language Toolkit — NLTK 3.2.5 documentation: 2017. *http://www.nltk.org/*. Accessed: 2017-12-14.

[8] TweetDreams: 2013. *https://ccrma.stanford.edu/groups/tweetdreams/*. Accessed: 2017-11-10.

[9] Twitter, Inc.: 2017. *https://github.com/twitter*. Accessed: 2017-12-14.