

## Haiku Twitter Bot

### Introduction:

The goal of Haiku Twitter Bot was to create a program, that would utilize the twitter api to find tweets whose syllable structures formed a haiku, then tweet these haikus in a formatted form, which evolved into tweeting a picture with the formatted haiku pasted into it. To do this we used the python module Tweepy to interact with the twitter api, the Natural Language Toolkit (NLTK) to separate the words of each tweet and count their syllables, and the python module Pillow (PIL) to provide functions used in pasting text into a picture. The way the program flows is by, first, using tweepy to create a twitter listener for people being followed by the Haiku Bot, or alternatively the public twitter stream, though this produces poor results due to the mass amounts of retweets that the program trips up on. After creating the twitter listener the program will wait for any new tweets, upon detecting one it will send the tweet text to a subroutine to start the process of detecting a haiku. The subroutine will first decide whether the tweet is in english and does not contain numbers, if either is false it will throw an exception and return to monitoring the twitter stream, if both are true it will remove all punctuation and separate the tweet into a list of words, then begin counting the syllables in each word. If the words fill out the correct syllable structure the subroutine will format the haiku into the 5,7,5 syllable lines and return the resulting string, this is all done using functions from the Natural Language Toolkit. After receiving the formatted tweet the program then passes it off to another subroutine which, using functions from the Pillow module, opens a stored image, in this case a photoshopped version of Matsuo Obashō,

then draws the text over the picture and saves the result, ending the subroutine. The listener then adds the screen name of the author of the tweet to the end of the haiku and tweets the haiku, as well as the modified picture and returns to monitoring the twitter stream.

### Project Results:

Bot account: <https://twitter.com/HaikusFromUForU?lang=en>

The bot accomplished everything we wanted it to do; get tweets from a list of users that the bot follows, detect if the text in the tweet is able to be put into haiku format, put the text into an image using python, and then finally tweeting the haiku and the image to twitter while crediting the original user. The ability to import python packages to help do things, such as, get twitter data, detect syllables, and manipulate images is very useful and it has become very obvious why python is so versatile and widely used.

A big pitfall we encountered was having inconsistent working environments. For one reason or another, Robbie's environment was inconsistent with Chris's, so the program would work on one's computer, while on the other it wouldn't work. The way we got around this was creating a virtual environment and making sure the exact same packages were installed. Another pitfall we overcame was listening to the wrong stream. First the program listened to the entire twitter public stream, but after many attempts, we were able to narrow it down to the bot's followed accounts by retrieving those accounts as a list and filtering through the public stream to find only tweets from those ids. Along

with the pitfalls, we had many small successes along the testing process like being able to tweet from python, finding a random haiku in the public stream and tweeting it out, and having the final product work as desired.

When running the program (python3 main.py), the command window will output every time it retrieves a tweet and show when it finds a haiku, tweet the haiku, and return to listening.

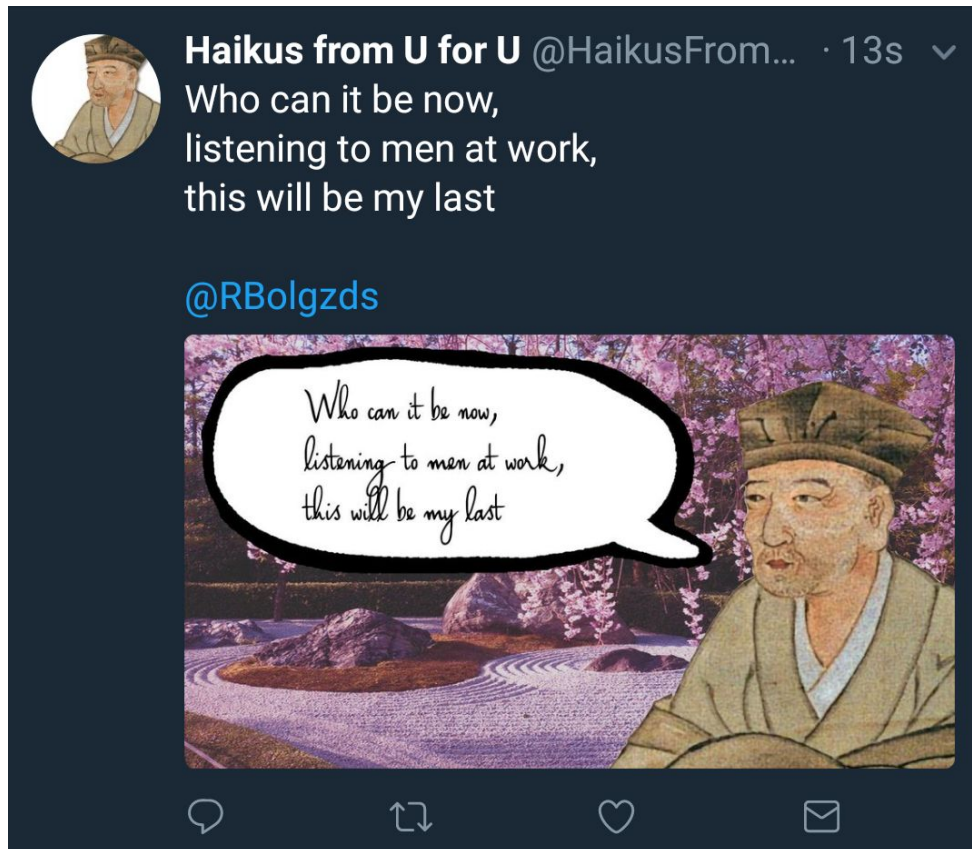
```
Chris@IDEAPAD-Y700:/mnt/c/Users/Chris_Messmer/Documents/CS2021-Final-Project$ python3 main.py
Stream established.
Found a tweet.
Found a tweet.
Found a tweet.
Found a tweet.
Found a tweet.
Found a tweet.
Found a tweet.
Found a tweet.
Found a tweet.
Found a tweet.

Found a haiku!

Who can it be now,
listening to men at work,
this will be my last

Haiku Tweeted!
```

In order to show that the program actually works Robbie tweeted haikus for the bot to find since haiku tweets are few and far between. But this does not mean that it only works for his account because it found a random user in the public stream and tweeted it out (with the wrong image) when we were still in the process of testing. This could be an idea for a potential improvement as the bot as it is now is not following a large number of people so it doesn't find haikus very often. The final tweet will look like this:



#### Division of Group Work:

The project was divided fairly evenly by the main parts of the project. Chris did the code related to detecting the haikus using the NLTK package and editing the image to tweet using the Pillow package. Robbie did the code related to pulling and posting from the twitter api using the tweepy package, as well as photoshopping the image used to tweet. Although there was a clear divide in each other's work, there were multiple instances where we helped each other with their portions. A couple of examples would be with the image editor program, it was mostly created by Chris and Robbie tweaked it to work with the photo template. Another example would be the streaming method

because we kept switching between different methods until Chris implemented the final stream object. The paper was evenly divided between both members evenly.

- Robbie's Contributions: My contributions mostly consisted of anything dealing with the twitter api. I also created the git repository, the bot's twitter account, and created the image tweeted out with the found haiku. I used the api object that we used in Lab 7 in class to establish a connection to our twitter bot, added the function to get all of the followed user's ids to only test tweets from followed accounts, and did the parts in the listener that dealt with twitter data, where Chris did the parts relating to the actual haiku.
- Chris's Contributions: My individual contribution was adding the syllable detection, which was mostly derived from the reference, though i did make some tweaks to make it easier to run on our machines, notably bypassing the use of the curses module. I also contributed to the creation of the listener and text pasting on the image along with tweeting these, though theses were really a joint effort between robbie and myself. Between all of these parts, I feel that we distributed the work on this project very evenly.

#### Bibliography:

*API Reference — tweepy 3.5.0 documentation.* (2017). *Docs.tweepy.org*. Retrieved 2 December 2017, from <http://docs.tweepy.org/en/v3.5.0/api.html>

Bird, Steven, Edward Loper and Ewan Klein (2009).

Natural Language Processing with Python. O'Reilly Media Inc.

*Haikus Script Font* | *dafont.com*. (2017). *Dafont.com*. Retrieved 2 December 2017, from <https://www.dafont.com/haikus-script.font>

*Streaming With Tweepy — tweepy 3.3.0 documentation*. (2017). *Docs.tweepy.org*. Retrieved 2 December 2017, from [http://docs.tweepy.org/en/v3.4.0/streaming\\_how\\_to.html](http://docs.tweepy.org/en/v3.4.0/streaming_how_to.html)

string, D. (2017). *Detect numbers in string*. *Stackoverflow.com*. Retrieved 2 December 2017, from <https://stackoverflow.com/questions/6649096/detect-numbers-in-string>

tweepy, F. (2017). *Full list of twitter "friends" using python and tweepy*. *Stackoverflow.com*. Retrieved 2 December 2017, from <https://stackoverflow.com/questions/25944037/full-list-of-twitter-friends-using-python-and-tweepy>

tweepy, R. (2017). *Retrieve tweets tweepy*. *Stackoverflow.com*. Retrieved 2 December 2017, from <https://stackoverflow.com/questions/41602826/retrieve-tweets-tweepy>

users, T. (2017). *Tweepy tracking terms and following users*. *Stackoverflow.com*. Retrieved 2 December 2017, from <https://stackoverflow.com/questions/14083968/tweepy-tracking-terms-and-following-use>

[rs](#)

*Using Python to Detect Haikus in the Public Twitter Stream - h6o6.* (2013). h6o6.

Retrieved 2 December 2017, from

<http://h6o6.com/2013/03/using-python-and-the-nltk-to-find-haikus-in-the-public-twitter-stream/>

tweepy, F. (2017). *Full list of twitter "friends" using python and tweepy.*

*Stackoverflow.com.* Retrieved 2 December 2017, from

<https://stackoverflow.com/questions/25944037/full-list-of-twitter-friends-using-python-and-tweepy>

Images used:

- poet  
<https://www.japanpowered.com/japan-culture/the-life-and-impact-of-matsuo-basho>
- Landscape  
<http://www.baroquestyle.com/vacation-destinations/2017/1/17/travel-guide-experience-cherry-blossom-season-in-japan>

Python libraries used:

- Tweepy - Twitter api interactions  
<https://github.com/tweepy/tweepy>
- Natural Language Toolkit (NLTK) - Language interpretation and manipulation  
<https://github.com/nltk/nltk>
- Pillow (PIL) - Image manipulation, text pasting  
<https://github.com/python-pillow/Pillow>

Code Appendix:

Github repository: <https://github.com/bolgzdrt/CS2021-Final-Project/tree/master>