TANGO

IA SENTIMENT MP3 PLAYER

TUTORIAL GUIDE

RONI BANDINI

IMPORTANT

THE AUTHOR MAKES NO REPRESENTATIONS OR WARRANTIES AS TO THE TIMELINESS, AVAILABILITY, ACCURACY OR COMPLETENESS OF ANY INFORMATION CONTAINED IN THIS DOCUMENT. THE INFORMATION PROVIDED IN THIS PDF MAY BECOME OUTDATED OVER TIME. THE AUTHOR UNDERTAKE NO OBLIGATION TO CORRECT OR UPDATE ANY CONTENT OR INFORMATION ON THIS DOCUMENT. THE AUTHOR MAY MAKE IMPROVEMENTS AND/OR CHANGES IN THE CONTENT AND INFORMATION AT ANY TIME WITHOUT NOTICE, AND RESERVE THE RIGHT TO ALTER OR DELETE CONTENT AND INFORMATION ON THE PDF AT ANY TIME. ALL INFORMATION ON THIS DOCUMENT IS PROVIDED “AS IS,” WITHOUT ANY WARRANTIES OF ANY SORT WHATSOEVER, ALL OF WHICH ARE EXPRESSLY DISCLAIMED.

INTRO

One of my childhood memories is my grandfather Vittorio listening music according to his mood. The most heart breaking tangos for sad days and milongas (happier and faster tangos) for sunny days.

Today we post our ideas, fears, hopes and feelings in Social Media. So, what about an Arduino based music player that determines your mood using artificial intelligence algorithms over your Tweets and automatically selects which Tangos to play?

SENTIMENT ANALYSIS

Sentiment analysis (also known as opinion mining or emotion AI) use natural language processing, text analysis, computational linguistics, and biometrics to identify, extract, quantify, and study affective states.

A common task in sentiment analysis is classifying the polarity of a given text as positive, negative, or neutral, which may seem simple for a phrase like “I’m feeling good today” but a little more challenging for something like “Hurricane again? What a great season!” (ironic)

Example: the tweet “A little sun, a good book… and a Dry Martini” over the PHP sentiment analysis library used in this project gets 0.304 positive score and “Oh boy… I’m tired of hurricanes” gets 0.367 negative score.

PARTS

For this project the following parts are used:

- ESP8266
- DfPlayer Mini
- 3 Leds
- Switch

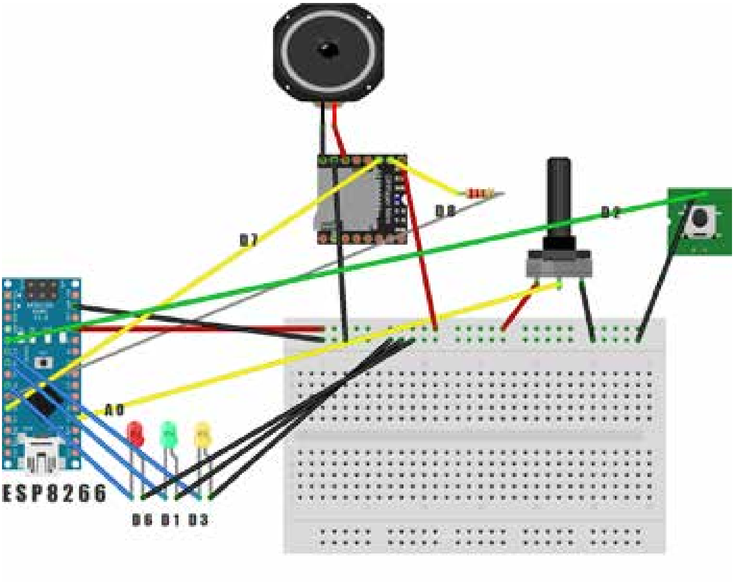
- Button
- Slider potentiometer
- 4ohm 3 watts speaker
- microSD card
- Custom 3d printed case

CIRCUIT

- DfPlayer mini to ESP D7 and D8 (rx, tx)
- DfPlayer mini to Speaker + and -
- Leds to D6 (red), D1 (green) and D3 (yellow) and GND
- Button to D2 and GND
- Potentiometer to A0, 5V and GND
- Switch between Power Supply 5V and VIN
- Power Supply GND to ESP GND

HOW DOES IT WORK

As soon as the device is turned on, a simple scan is made to verify that all Leds are working.



Then, WiFi led will blink until a connection to local router is established.

At this point a query is made to a remote Apache Server running Twitter API and PHP Sentiment Analyzer library. The latest 5 tweets of a user timeline will be considered and a simple "h" will be returned for happy sentiment or "s" for sad sentiment.

The device will parse server response and turn on the green Led (happy) or red Led (sad) It will also point to respective folder (happy tangos in folder 01 and sad tangos in folder 02)

The press of the button will start playing random Tangos in selected folder. The same button will pause. The potentiometer slider is used to control DFPlayer mini digital volume (from 1 to 30)

SERVER SIDE SETUP

You will need Twitter API credentials. A tutorial can be found here https://developer.twitter.com/en/docs/authentication/oauth-1-0a/obtaining-user-access-tokens

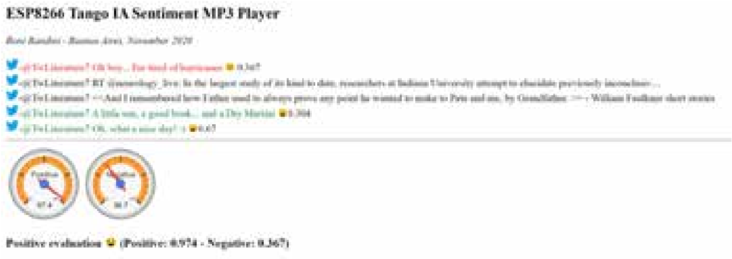
Then you need to add credentials to PHP

code

```
$oauth_access_token = "";
$oauth_access_token_secret = "";
$consumer_key = "";
$consumer_secret = "";
```

Add also your Twitter account – without @ - in this line:

```
'screen_name'      => 'YourTwitterScreenName',
```



ESP8266 Tango IA Sentiment MP3 Player

Roni Bandini - Buenos Aires, November 2020

@TwLiterature7 Oh boy... The kind of hurricane ● 0.367
@TwLiterature7 RT @neurology_live: In the largest study of its kind to date, researchers at Indiana University attempt to elucidate previously inconclusive ...
@TwLiterature7 <<And I remembered how Father used to always prove any point he wanted to make to Pete and me, by Grandfather >> - William Faulkner short stories
@TwLiterature7 A little sun, a good book... and a Dry Martini ● 0.394
@TwLiterature7 Oh, what a nice day! :) ● 0.67

Positive evaluation 🙂 (Positive: 0.974 - Negative: 0.367)

You will also need to install this PHP library in your server https://github.com/davmixcool/php-sentiment-analyzer using

```
composer require davmixcool/php-sentiment-analyzer
```

Finally, edit your server domain inside Arduino code

```
const String serverCall=”http://your server here.com/sent.php?debug=0”;
```

In order to test Twitter API integration and sentiment analysis library, you can manually call the PHP script with the debug setting. Example:

```
http://yourServer.com/tango/sent.php?debug=1
```

CODE

ARDUINO

```
/*
Tango Sentiment Analysis Player
Roni Bandini - Buenos Aires 2020
@RoniBandini
*/

#include “SoftwareSerial.h”
#include <ESP8266WiFi.h>
#include <ESP8266HTTPClient.h>
#include <DHT.h>

const char* ssid = “YourSSIDHere”;
const char* password = “YourWIFIPasswordHere”;
const String serverCall=”http://YourDomainHere.com/sent.
```

```
php?debug=0”;

// Mp3 player rx and tx
SoftwareSerial mySerial(D7, D8);

# define Start_Byte 0x7E
# define Version_Byte 0xFF
# define Command_Length 0x06
# define End_Byte 0xEF
# define Acknowledge 0x00
# define ACTIVATED LOW

int potPin = 0;
int butPin = D2;

// leds
int ledRed = D6;
int ledGreen = D1;
int ledWiFi=D3;

int myVolume=20;
int isPlaying=0;

long randNumber;

String sentiment=”-”;

// init array
int arrayCounter=0;
int arrayUsed[25] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0};
int songPlayed=1;

void setup(){
```

```
// mp3 player
mySerial.begin(9600);

Serial.begin(9600);
Serial.println(“Tango Sentiment Player started”);

// nothing being played, just init
playFirst();

// button
pinMode(butPin, INPUT_PULLUP);

// leds
pinMode(ledRed, OUTPUT);
pinMode(ledGreen, OUTPUT);
pinMode(ledWiFi, OUTPUT);

// test leds

digitalWrite(ledWiFi, HIGH);
delay(100);
digitalWrite(ledWiFi, LOW);
digitalWrite(ledRed, HIGH);
delay(100);
digitalWrite(ledRed, LOW);
digitalWrite(ledGreen, HIGH);
delay(100);
digitalWrite(ledGreen, LOW);

WiFi.begin(ssid, password);

int ledOn=0;
```

```
    while (WiFi.status() != WL_CONNECTED) {

       delay(500);
       Serial.print(“Connecting..”);

       if (ledOn==0){
        digitalWrite(ledWiFi, HIGH);
        ledOn=1;
       }
       else
       {
        digitalWrite(ledWiFi, LOW);
        ledOn=0;
        }
    }

    // set volume
    int sliderValue = map(analogRead(potPin), 0, 1023, 0,
30);

    myVolume=sliderValue;
    setVolume(myVolume);

   if (WiFi.status() == WL_CONNECTED) { //Check WiFi
connection status


            HTTPClient http;

            http.begin(serverCall);
            int httpCode = http.GET();

            if (httpCode > 0) {
                   digitalWrite(ledWiFi, HIGH);

                   String payload = http.getString();
                   Serial.println(payload);

                    // parse server response
                    sentiment        = getValue(payload,
'#', 1);
                    Serial.println(" - Sentiment:
"+sentiment);



                    if (sentiment=="h"){

                       digitalWrite(ledGreen, HIGH);
                       digitalWrite(ledRed, LOW);
                       Serial.print("Hapy");




                    }
                   else
                    {
                     digitalWrite(ledGreen, LOW);
                     digitalWrite(ledRed, HIGH);
                     Serial.print("Sad");

                    }




                    } // http ok
```

```
                                            else
                                            {
                                              Serial.print(" - http error");
                                             }

                                           http.end();

                                           }

}


void loop(){

    int butValue = digitalRead(butPin);
    int potValue = analogRead(potPin);
    int sliderValue = map(potValue, 0, 1023, 0, 30);

    // tolerance of 2 to avoid noise
    if (sliderValue<myVolume-2 or sliderValue>myVolume+2){

      Serial.println("Changing volume");
      Serial.println(sliderValue);

      myVolume=sliderValue;
      setVolume(myVolume);

      }

    if (butValue==0){

         // button pressed

         if (isPlaying==0){
}


       if (sentiment==”h”){
        songPlayed=selectSound(1,9);
        Serial.println(“Playing Happy: “);
        Serial.println(songPlayed);
        execute_CMD(0x0F,0x01,(byte)songPlayed);
        //execute_CMD(0x08, 0, 0);
       }
       else
       {
        songPlayed=selectSound(1,7);
        Serial.println(“Playing Sad: “);
        Serial.println(songPlayed);
        execute_CMD(0x0F,0x02,(byte)songPlayed);
        }
      isPlaying=1;
     }
     else
     {
      pause();
      isPlaying=0;
      }

      delay(200);


     } // but pressed

    delay(25);

}


void playFirst()
```

```
{
  execute_CMD(0x3F, 0, 0);
  delay(500);
  setVolume(25);
  delay(500);

}

void pause()
{
  execute_CMD(0x0E,0,0);
  delay(500);
}

void play()
{
  execute_CMD(0x0D,0,1);
  delay(500);
}

void playNext()
{
  execute_CMD(0x01,0,1);
  delay(500);
}

void playPrevious()
{
  execute_CMD(0x02,0,1);
  delay(500);
}
```

```
void setVolume(int volume)
{
  execute_CMD(0x06, 0, volume); // Set the volume
(0x00~0x30)
  delay(2000);
}

void execute_CMD(byte CMD, byte Par1, byte Par2)
// Excecute the command and parameters
{
   // Calculate the checksum (2 bytes)
   word checksum = -(Version_Byte + Command_Length +
CMD + Acknowledge + Par1 + Par2);
   // Build the command line
   byte Command_line[10] = { Start_Byte, Version_Byte,
Command_Length, CMD, Acknowledge,
   Par1, Par2, highByte(checksum), lowByte(checksum),
End_Byte};
   //Send the command line to the module
   for (byte k=0; k<10; k++)
   {
     mySerial.write( Command_line[k]);
   }
}

int selectSound(int lowerLimit, int upperLimit)
{
  //Serial.println("Selecting sound from :"+String(lowerLimit)+
" to " + String(upperLimit));

  // check if all were assigned
  int oneEmpty=0;
  for (byte i = lowerLimit; i < upperLimit; i = i + 1) {
    if (arrayUsed[i]==0){
      oneEmpty=1;
      }
  }// for

  if (oneEmpty==0){
    // no empty spots, clean all
    Serial.println("Cleaning array");
    for (byte i = lowerLimit; i < upperLimit; i = i + 1) {
      arrayUsed[i]=0;
      }
    }// if

  int foundSpot=0;

  while (foundSpot==0) {

      randNumber = random(lowerLimit, upperLimit);
      //Serial.println("Trying "+ String(randNumber));
      if (arrayUsed[randNumber]==0) {
          arrayUsed[randNumber]=1;
          //Serial.println("OK");
          foundSpot=1;
       }

    }

  Serial.println("Selected :"+String(randNumber));
  return randNumber;

} // end loop

String getValue(String data, char separator, int index)
{
  int found = 0;
  int strIndex[] = {0, -1};
  int maxIndex = data.length()-1;

  for(int i=0; i<=maxIndex && found<=index; i++){
    if(data.charAt(i)==separator || i==maxIndex){
        found++;
        strIndex[0] = strIndex[1]+1;
        strIndex[1] = (i == maxIndex) ? i+1 : i;
    }
  }

  return found>index ? data.substring(strIndex[0],
strIndex[1]) : "";
}
```

PHP SERVER SIDE

```
<?php

// Tango IA Sentiment MP3 Player
// Roni Bandini
// Buenos Aires, Argentina 2020
// @RoniBandini
// sent.php

$debug                    =$_GET["debug"];
$screen_name              =$_GET["screen_name"];

require_once("vendor/autoload.php");

Use Sentiment\Analyzer;
```

```
$sentiment = new Sentiment\Analyzer();

$strings[0]=returnTweet()[0][‘full_text’];
$strings[1]=returnTweet()[1][‘full_text’];
$strings[2]=returnTweet()[2][‘full_text’];
$strings[3]=returnTweet()[3][‘full_text’];
$strings[4]=returnTweet()[4][‘full_text’];
$strings[5]=returnTweet()[5][‘full_text’];

// new words not in the dictionary
$newWords = [
            ‘rubbish’=> ‘-1.5’,
            ‘mediocre’ => ‘-1.0’,
            ‘agressive’ => ‘-0.5’
];

if ($debug==”1”){

?>
   <script type=”text/javascript” src=”https://www.gstatic.
com/charts/loader.js”></script>

  </head>

  <body>

  <h2>Tango IA Sentiment MP3 Player</h2>
  <i>Roni Bandini - Buenos Aires, November 2020</i><br>
  </body>
  <?



}
```

```
//Dynamically update the dictionary with the new words
$sentiment->updateLexicon($newWords);

$positiveCounter=0;
$negativeCounter=0;
$neutralCounter=0;

//Print results
foreach ($strings as $string) {

        if ($string!=””){
                        // calculations:
                        $scores = $sentiment-
>getSentiment($string);

                        //print_r(json_encode($scores));

                        if ($scores[‘compound’]>0){
                                if ($debug==”1”){
                                        echo “<br><img
src=’images/twitter.png’ width=20><font color=green>”;
                                        echo “-@”.$screen_
name.” “.$string.”\n”;
                                        echo “</font>”;
                                        echo
“&#128516;”.$scores[‘pos’];
                                }

//$positiveCounter=$positiveCounter+$scores[‘pos’];

                        }
                        if ($scores[‘compound’]<0)
                        {
                                if ($debug==”1”){
```

```
                    echo “<br><img
src=’images/twitter.png’ width=20><font color=red>”;
                    echo “-@”.$screen_
name.” “.$string.”\n”;
                    echo “</font>”;
                    echo “&#128528;
“.$scores[‘neg’];
                }

$negativeCounter=$negativeCounter+$scores[‘neg’];
            }
            if ($scores[‘compound’]==0)
            {
                if ($debug==”1”){
                    echo “<br><img
src=’images/twitter.png’ width=20>-@”.$screen_name.”
“.$string.”\n”;
                }

$neutralCounter=$neutralCounter+$scores[‘neu’];
            }
            //print(“Comp:
“.$scores[‘compound’]);
    }

}


if ($debug==”1”){
    ?>
<script type=”text/javascript”>
  google.charts.load(‘current’, {‘packages’:[‘gauge’]});
  google.charts.setOnLoadCallback(drawChart);
```

```
    function drawChart() {

     var data = google.visualization.arrayToDataTable([
       ['Label', 'Value'],
       ['Positive', <?=$positiveCounter*100?>],
       ['Negative', <?=$negativeCounter*100?>]
     ]);

     var options = {
       width: 400, height: 120,
       redFrom: 0, redTo: 0,
       yellowFrom:0, yellowTo: 100,
       minorTicks: 5
     };

                     var options2 = {
       width: 400, height: 120,
       redFrom: 0, redTo: 100,
       yellowFrom:0, yellowTo: 0,
       minorTicks: 5
     };

     var chart = new google.visualization.Gauge(document.
getElementById('chart_div'));

     chart.draw(data, options);

     setInterval(function() {
       data.setValue(0, <?=$positiveCounter*100?>,
<?=$positiveCounter*100?> + Math.round(5 * Math.
random()));
       chart.draw(data, options);
     }, 13000);
     setInterval(function() {
        data.setValue(1, <?=$negativeCounter*100?>,
<?=$negativeCounter*100?> + Math.round(5 * Math.
random()));
        chart.draw(data, options2);
      }, 5000);
    }
   </script>
           <hr>
   <div id=”chart_div” style=”width: 400px; height: 120px;”></
div>
<?
}

if ($positiveCounter>$negativeCounter){

           if ($debug==”1”){
                      echo “<h3>Positive evaluation
&#128516; (Positive: “.$positiveCounter.” - Negative:
“.$negativeCounter.”)</h3>”;
           }
           else echo “#h”;
}
else
{
           if ($debug==”1”){
                      echo “<h3>Negative evaluation &#128528;
(Positive: “.$positiveCounter.” - Negative: “.$negativeCounter.
“)</h3>”;
           }
           else echo “#s”;
}


function buildBaseString($baseURI, $method, $params) {
    $r = array();
    ksort($params);
    foreach($params as $key=>$value){
        $r[] = “$key=” . rawurlencode($value);
    }
    return $method.”&” . rawurlencode($baseURI) . ‘&’ .
rawurlencode(implode(‘&’, $r));
}

function buildAuthorizationHeader($oauth) {
    $r = ‘Authorization: OAuth ‘;
    $values = array();
    foreach($oauth as $key=>$value)
        $values[] = “$key=\”” . rawurlencode($value) . “\””;
    $r .= implode(‘, ‘, $values);
    return $r;
}

function returnTweet(){

    $oauth_access_token = “”;
    $oauth_access_token_secret = “”;
    $consumer_key = “”;
    $consumer_secret = “”;



    $twitter_timeline           = “user_timeline”;  //  mentions_
timeline / user_timeline / home_timeline / retweets_of_me

    //  create request
        $request = array(
            ‘screen_name’       => ‘TwLiterature’,
                              'tweet_mode'           =>
'extended',
                              'SearchParams'         =>
'exclude:retweets',
      'count'         => '5'
    );

  $oauth = array(
    'oauth_consumer_key'      => $consumer_key,
    'oauth_nonce'             => time(),
    'oauth_signature_method'  => 'HMAC-SHA1',
    'oauth_token'             => $oauth_access_token,
    'oauth_timestamp'         => time(),
    'oauth_version'           => '1.0'
  );

  //  merge request and oauth to one array
    $oauth = array_merge($oauth, $request);

  //  do some magic
    $base_info          = buildBaseString("https://api.twitter.
com/1.1/statuses/$twitter_timeline.json", 'GET', $oauth);
    $composite_key      = rawurlencode($consumer_
secret) . '&' . rawurlencode($oauth_access_token_secret);
    $oauth_signature        = base64_encode(hash_
hmac('sha1', $base_info, $composite_key, true));
    $oauth['oauth_signature']   = $oauth_signature;

  //  make request
    $header = array(buildAuthorizationHeader($oauth),
'Expect:');
    $options = array( CURLOPT_HTTPHEADER => $header,
              CURLOPT_HEADER => false,
              CURLOPT_URL => "https://api.twitter.
```

```
com/1.1/statuses/$twitter_timeline.json?”. http_build_
query($request),
                    CURLOPT_RETURNTRANSFER => true,
                    CURLOPT_SSL_VERIFYPEER => false);

      $feed = curl_init();
      curl_setopt_array($feed, $options);
      $json = curl_exec($feed);
      curl_close($feed);

   return json_decode($json, true);
}
?>
```

WHERE TO GO FROM HERE

In this version, IA Sentiment Analysis is used for Social Media but not for music. Music is previously categorized and placed into folders. A nice modification will be to apply the same IA algorithm to music lyrics.

ELECTRONIC ART PROJECTS

- Ted Talk about Arduino devices

- Jorge Luis Borges Animatronic with a hacked Furby
- The Klausner Machine
- Song recorded with Arduino instruments
- Mind Poetry
- BookSound
- Literature dispenser
- Rayuelomatic

CONTACT

If you need assistance please use comments area of this project

Instagram @RoniBandini
Twitter @RoniBandini

Roni Bandini, Buenos Aires, 2020