



GHOSTWRITER

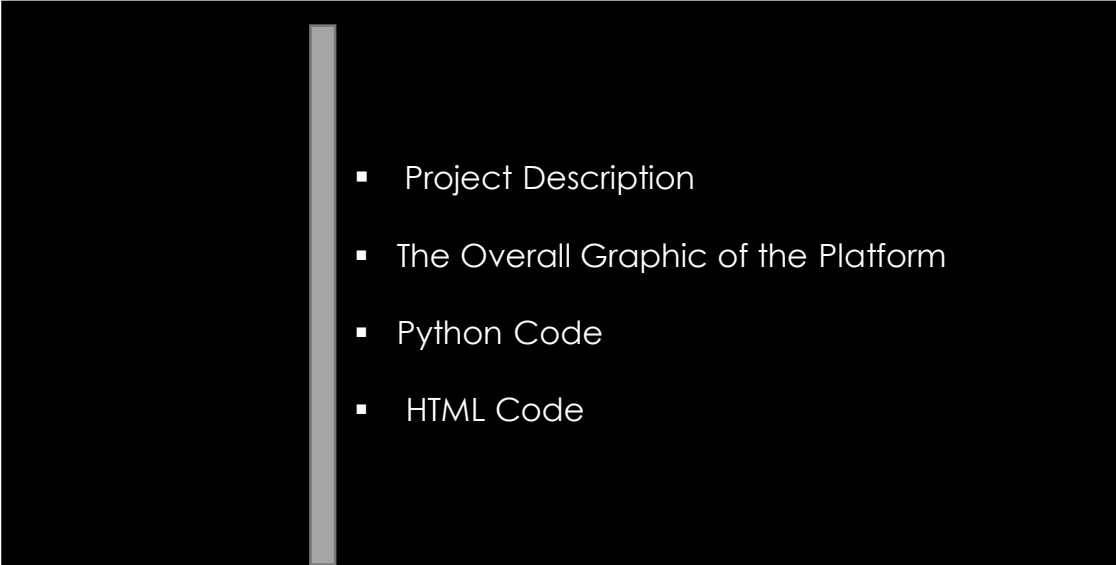
Author:

Zeinab Rahimi

Supervisor:

Dipl.-Designer Mattis Kuhn

CONTENT :

- 
- Project Description
 - The Overall Graphic of the Platform
 - Python Code
 - HTML Code

Let's Write Positive

Zeinab Rahimi



Project Description

“Let’s write positive” is designed based on the idea that the world could be a little bit more positive place to live in.

The idea of the project was inspired by reading negative comments on social networks. The absence of eye contact or other in-person interactions in virtual environments, such as social networks, gives people the right to express negative words against each other. By showing the writer the negativity score of a text, we hope to encourage society to use more positive language when writing a text. The tool can be used in any other context, for instance, in writing a digital diary that our grandchildren will eventually find and read.

Let's Write Positive

Zeinab Rahimi



Technical description

In this project, sentiment analysis will be applied to a piece of text that the visitor types using a keyboard. A sentiment analysis tool would analyze the text, and the result would be displayed on the screen for the visitor to see; if the result indicates a certain level of negativity, the program would either use the internal pre-defined dictionary to alter the user's input to a more positive one or deletes the negative text.

The dictionary of antonym words is defined by the author, so the negative words would be replaced by their defined antonym.

The project is written in the Python programming language and the NLTK sentiment analysis library is used to perform the sentiment analysis.

Users can interact with the program via a web-based platform created with the Flask framework.

User Interface

Let's WRITE

+

P

OSitive

Type Your Text Here . . .

Your Text Scores:

00.00

00.00

00.00

Negativity

Positivity

Neutrality

Let's Write Positive

Zeinab Rahimi

Python Code:

```
dictionary = {'disgusting': 'fantastic', 'awful': 'excellent', 'hate': 'love', 'hateful': 'wonderful',
              'weird': 'normal', 'arrogant': 'humble', 'undesirable': 'desirable', 'heartless': 'compassionate',
              'insane': 'rational', 'messy': 'organized', 'bad': 'good', 'worst': 'best', 'ignorant': 'intelligent',
              'narcissistic': 'unselfish', 'rude': 'polite', 'gross': 'great', 'brutal': 'gentle', 'cheap': 'admirable',
              'stupid': 'clever', 'ugly': 'beautiful', 'unvaluable': 'valuable', 'boring': 'interesting', 'cruel': 'compassionate',
              'dishonest': 'honest', 'harsh': 'pleasant', 'jealous': 'trustful', 'narrow-minded': 'open-minded',
              'untrustworthy': 'trustworthy', 'unhappy': 'happy', 'impatient': 'patient', 'selfish': 'generous',
              'egotistic': 'modest', 'silly': 'thoughtful', 'moody': 'stable', 'coward': 'fearless', 'impolite': 'polite',
              'aloof': 'boon', 'bigoted': 'broad-minded', 'fussy': 'patient', 'grumpy': 'good-tempered', 'violent': 'calm',
              'antisocial': 'social', 'naive': 'experienced', 'obnoxious': 'acceptable', 'possessive': 'permissive',
              'sadistic': 'compassionate', 'manipulative': 'honest', 'hostile': 'friendly'}
```

```
from flask import Flask, render_template, request, jsonify
from nltk.sentiment import SentimentIntensityAnalyzer as SIA
sia = SIA()

app = Flask(__name__)

# Show main page
@app.route('/')
def index():

    return render_template('index.html')

# Process data which was posted through js
@app.route('/process', methods=['POST'])
def process():
    # Store the user input
    data = request.form['user_input']

    # Store score of the sentiment analysis
    scores_old = sia.polarity_scores(data)

    # Process the input
    data=Positive_Maker(data)

    # Sentiment Analysis
    scores_new = sia.polarity_scores(data)

    # Send the modified text + its scores back to the js script.
    return jsonify({
        'output': data,
        'neg_old': scores_old['neg'],
        'pos_old': scores_old['pos'],
        'neu_old': scores_old['neu'],
        'neg_new': scores_new['neg'],
        'pos_new': scores_new['pos'],
        'neu_new': scores_new['neu']
    })

def change_words(txt, dictionary):
    return " ".join([dictionary.get(w, w) for w in word_tokenize(txt)])

def Positive_Maker(data):
    txt = data
    sia = SIA()
    debug = True # Set this to True if you want to see intermediate steps of the text processing
    scores = sia.polarity_scores(txt)

    ### Step 1: Remove negations ###
    if scores['neg'] > 0.3:
        if debug: print('Input in step 1:', txt)

        # Optional: Store input in a text file on your disk
        save_negative_sentence(txt)

        # Change words like don't and doesn't before the tokenization happens
        txt = txt.replace('not', '')
        txt = txt.replace("don't", '')
        txt = txt.replace("n't", '')

        txt= txt.replace(' ', ' ') # Remove double spaces

        scores = sia.polarity_scores(txt)

        if debug: print('Output of step 1:', txt, '\n')

    ### Step 2: Change words ###
    # Check if step 1 is enough
    # Otherwise change words
    if scores['neg'] > 0.3:
        if debug: print('Input in step 2:', txt)

        txt = change_words(txt, dictionary)
        txt= re.sub(r"\'s(?:[?!,.])(?:\'s|$)", r'\1', txt)
        scores = sia.polarity_scores(txt)

        if debug: print('Output of step 2:', txt, '\n')

    ### Step 3: Remove whole sentences if necessary ###
    # Check if step 2 is enough,
    # otherwise analyze individual sentences and remove bad ones entirely.
    if scores['neg'] > 0.3:
        if debug: print('Input in step 3:', txt)

        # Split txt into sentences
        sentences = sent_tokenize(txt)
        cleaned_txt = [] # A list to store the positive sentences
        for sentence in sentences:
            scores = sia.polarity_scores(sentence)
            # Append sentence to cleaned_txt if the score is below a threshold
            if scores['neg'] < 0.3:
                cleaned_txt.append(sentence)

        # Join cleaned_txt
        txt = ' '.join(cleaned_txt)

        if debug: print('Output of step 3:', txt, '\n')

    # Return modified text
    return txt

# This function appends a sentence to a (daily) text file
# It can be used in operation to save the user inputs for
# further improvement etc.
def save_negative_sentence(sentence):
    from pathlib import Path
    from datetime import datetime
    date = datetime.now().strftime('%Y_%m_%d') # Like 2022_08_23

    # Path for writing the file
    source_path = Path(__file__).resolve()
    source_dir = str(source_path.parent)
    # Append sentence to a file (creates the file if it does not exist)
    with open(source_dir + '/negative_inputs_' + date + '.txt', 'a') as f:
        f.write(sentence)
        f.write('\n')

if __name__ == '__main__':
    app.debug = True # Set this to False when the development is done.
    app.run(port=7000)
```

Let's Write Positive

Zeinab Rahimi

HTML Code :

```
index.html X
C: > Users > anahita > Dropbox > My PC (DESKTOP-QSING628) > Downloads > Goastwriter > project > august > flask > flask > templates > index.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <meta name="keywords" content="Comma-separated keywords">
7    <meta name="description" content="Brief description">
8    <meta name="author" content="You">
9    <title>Let's WRITE POSitive</title>
10
11    <link rel="stylesheet" href="{{ url_for('static', filename='flexboxgrid.min.css') }}">
12    <link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
13    <script src="{{ url_for('static', filename='jquery-3.4.1.min.js') }}" charset="utf-8"></script>
14  </head>
15  <body>
16
17    <div class="container">
18
19      <!-- LOGO start -->
20      <div class="row logo">
21        
22      </div>
23      <!-- LOGO end -->
24
25      <!-- CONTENT start -->
26      <div class="row content">
27
28        <!-- LEFT COL (INPUT) -->
29        <div class="col-xs-12 col-sm-6">
30          <form>
31            <textarea id = 'user_input' placeholder="Type Your Text Here (English)..."></textarea><br>
32          </form>
33        </div>
34
```

```

35
36      <!-- RIGHT COL (ANALYSIS) -->
37      <div class="col-xs-12 col-sm-6">
38        <div class="scores-headline">Your Text Scores:</div>
39
40        <div class="scores scores-red">
41          <div class="scores-value" id="scores_neg">
42            0.00
43          </div>
44
45          <div class="scores-label">
46            N<br>
47            egativity
48          </div>
49        </div>
50
51        <div class="scores scores-green">
52          <div class="scores-value" id="scores_pos">
53            0.00
54          </div>
55
56          <div class="scores-label">
57            P<br>
58            ositivity
59          </div>
60        </div>
61
62        <div class="scores scores-gray">
63          <div class="scores-value" id="scores_neu">
64            0.00
65          </div>
66
67          <div class="scores-label">
68            N<br>
69            eutrality
70          </div>
71        </div>
72
73      </div>
74
```

Let's Write Positive

Zeinab Rahimi

```
74
75 <!-- stripe -->
76 <hr>
77
78 </div>
79 <!-- CONTENT end -->
80
81 </div>
82
83 <script type="text/javascript">
84
85 // Reset scores when the textarea is empty
86
87 $(document).ready(function() {
88   $('#user_input').on('keyup', function() {
89     var key = event.keyCode || event.charCode;
90
91     if ( key == 8 ) {
92       if ($('#user_input').val().length == 0) {
93         $('#scores_neg').text("0.0");
94         $('#scores_pos').text("0.0");
95         $('#scores_neu').text("0.0");
96       }
97     }
98   });
99 });
100
```

```
103
104 $(document).ready(function() {
105
106   // Check user input on every keypress
107   $('#user_input').keypress(function( event ) {
108
109     // Event listener for ['.', '?', '!'] = [46, 63, 33]
110     // event.which holds the keycode (46 for . etc)
111     if ( [46, 63, 33].indexOf(event.which) >= 0 ) {
112
113       // Get the content from the textarea
114       var text = $('#user_input').val();
115       // Append the string of the current key ('.', '?', '!')
116       text += String.fromCharCode(event.which);
117
118       // Submit text to /process
119       $.ajax({
120         data : {
121           user_input : text
122         },
123         type : 'POST',
124         url : '/process'
125       },
126       .done(function(data) {
127         // data holds the data returned from processing
128         // Insert processed data (via id) into the html code
129         $('#scores_neg').text(data.i_neg);
130         $('#scores_pos').text(data.i_pos);
131         $('#scores_neu').text(data.i_neu);
132
133
134         // Timeout
135         setTimeout(function () {
136           $('#user_input').val(data.output);
137           $('#scores_neg').text(data.neg);
138           $('#scores_pos').text(data.pos);
139           $('#scores_neu').text(data.neu);
140
141         }, 1000); // value in milliseconds

```


Let's Write Positive

Zeinab Rahimi

```
151
152     var idleTime = 0;
153     var maxIdleTime = 30; // clear after 30 seconds
154     $(document).ready(function () {
155         // Increment the idle time counter every minute.
156         var idleInterval = setInterval(timerIncrement, 1000); // 1 second
157
158         // Zero the idle timer on mouse movement.
159         $(this).mousemove(function (e) {
160             idleTime = 0;
161         });
162         $(this).keypress(function (e) {
163             idleTime = 0;
164         });
165     });
166
167     function timerIncrement() {
168         idleTime = idleTime + 1;
169         if (idleTime >= maxIdleTime) {
170             $('#user_input').val('');
171             $('#scores_neg').text("0.0");
172             $('#scores_pos').text("0.0");
173             $('#scores_neu').text("0.0");
174         }
175     }
176
177
178 </script>
179 </body>
180 </html>
```

Visitor's interaction with the App at Ars Electronica 2022

