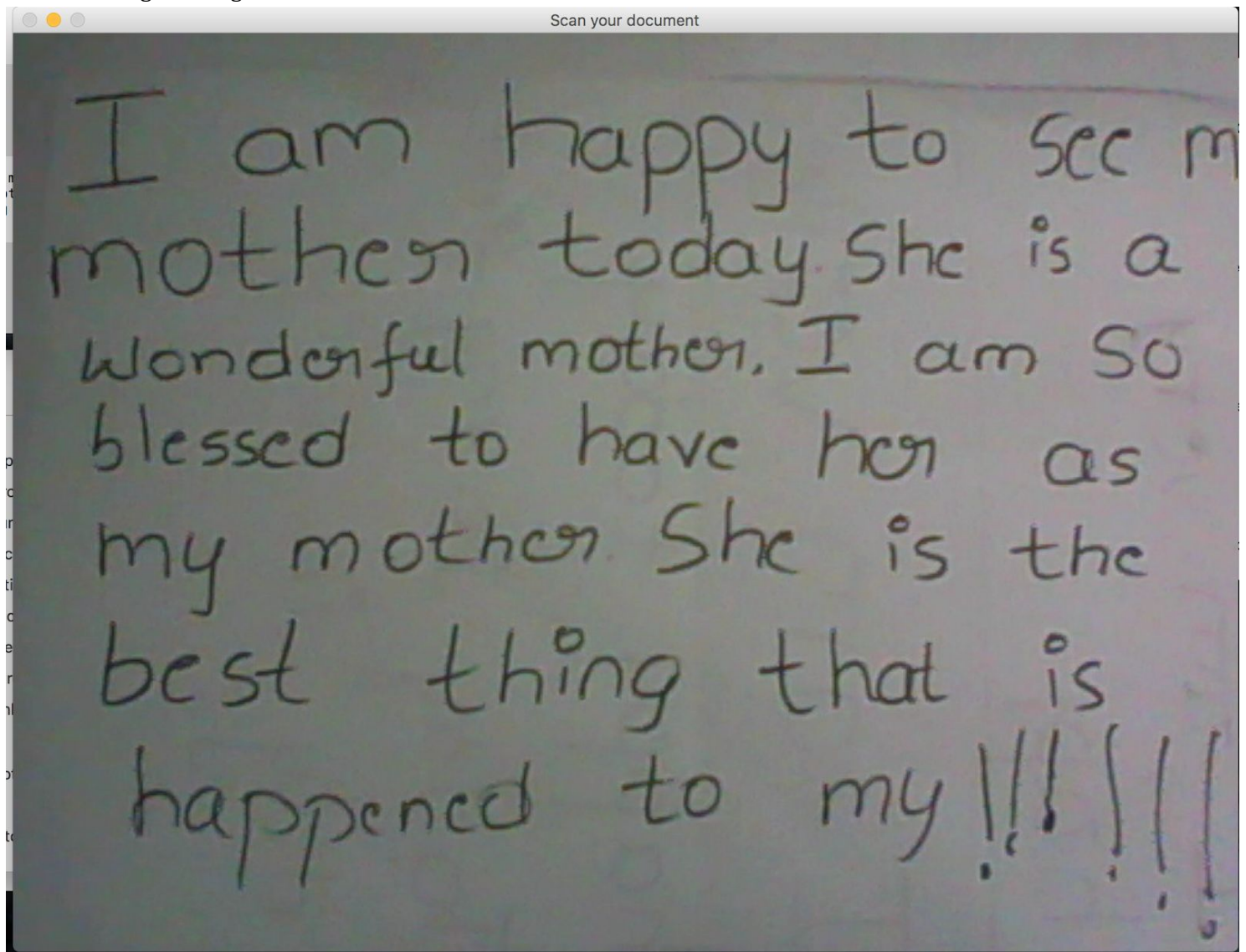


Output:

STEP 1:

Due to wide range input that exists in the environment . We have extended our system to accept two type of inputs . input can be voice or scanned digital image . Input of two types is showed below . You can use any one type of input both of with produces promising results , for this system we have considered scanned digital image.

1: scanned digital image

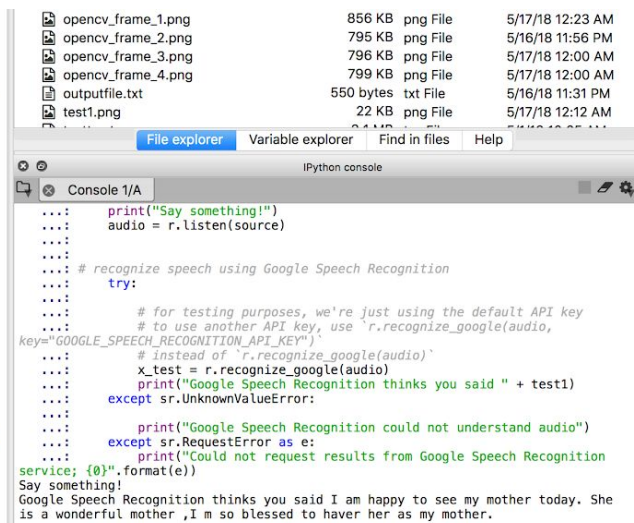


2:voice

```

136 '''
137 Input 1) Voice
138 '''
139 #
140 #
141 #
142 import speech_recognition as sr
143
144 # obtain audio from the microphone
145 r = sr.Recognizer()
146 with sr.Microphone() as source:
147
148     print("Say something!")
149     audio = r.listen(source)
150
151 # recognize speech using Google Speech Recognition
152 try:
153     # for testing purposes, we're just using the default API key
154     # to use another API key, use 'r.recognize_google(audio, key="GOOGLE_SPEECH_RECOGNITION_API_KEY")'
155     # instead of 'r.recognize_google(audio)'
156     x_test = r.recognize_google(audio)
157     print("Google Speech Recognition thinks you said " + x_test)
158 except sr.UnknownValueError:
159     print("Google Speech Recognition could not understand audio")
160 except sr.RequestError as e:
161     print("Could not request results from Google Speech Recognition service; {}".format(e))
162
163 # End of Speech recognition
164
165
166
167
168
169
170 '''
171 Emotion Detection
172 '''

```



The screenshot shows a Jupyter Notebook interface. At the top, there is a file explorer showing several files: opencv_frame_1.png (856 KB), opencv_frame_2.png (795 KB), opencv_frame_3.png (796 KB), opencv_frame_4.png (799 KB), outputfile.txt (550 bytes), and test1.png (22 KB). Below the file explorer is the variable explorer, which is currently empty. At the bottom is the IPython console, which displays the output of the code executed in the notebook. The console shows the following output:

```

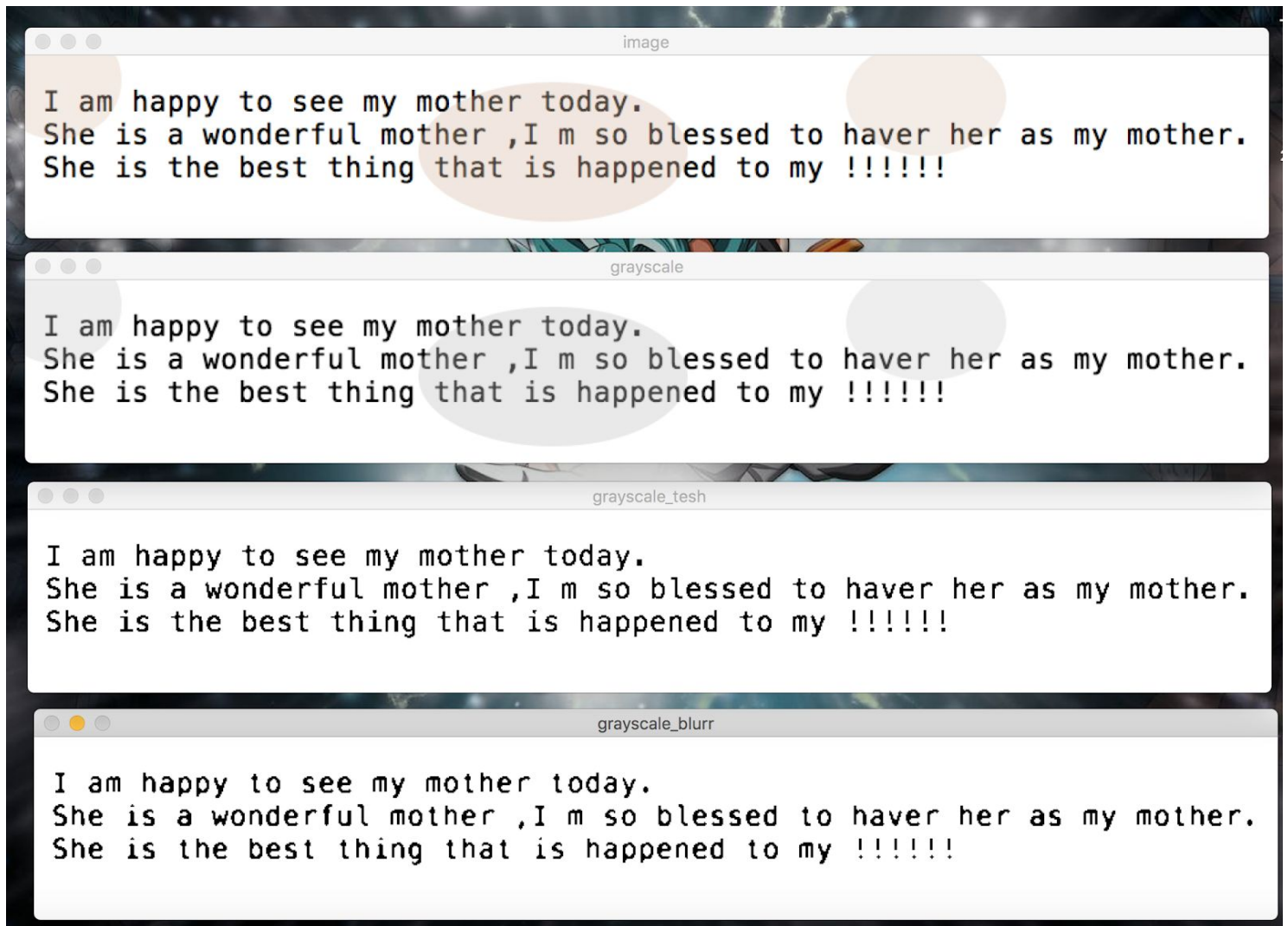
...: print("Say something!")
...: audio = r.listen(source)
...:
...: # recognize speech using Google Speech Recognition
...: try:
...:     # for testing purposes, we're just using the default API key
...:     # to use another API key, use 'r.recognize_google(audio,
...:     # instead of 'r.recognize_google(audio)'
...:     x_test = r.recognize_google(audio)
...:     print("Google Speech Recognition thinks you said " + test1)
...: except sr.UnknownValueError:
...:     print("Google Speech Recognition could not understand audio")
...: except sr.RequestError as e:
...:     print("Could not request results from Google Speech Recognition
service; {}".format(e))
Say something!
Google Speech Recognition thinks you said I am happy to see my mother today. She
is a wonderful mother ,I m so blessed to haver her as my mother.

```

STEP 3:

Usually images contains noise ,which causes unnecessary problem while processing the image.

The input scanned image which is given to the system is preprocessed so remove unnecessary noise in the image.



STEP 3:

After preprocessing the input scanned image to remove unnecessary noise. Preprocessed image is given to ocr model of the system . The model produces the answer which contains charactes / sentence encapsulated inside the image .Which is used for further processing in the system.

```

124 #
125 # =====
126 # Using Tesseract API to extract text from preprocessed image
127 # =====
128 text = pytesseract.image_to_string(Image.open(filename))
129 print(text)
130 os.remove(filename)
131
132 text=text.split()
133 test1=' '.join(text)
134
135
136 '''
137 Input 1) Voice
138 '''
139 # =====
140 # =====
141 # =====
142 import speech_recognition as sr
143
144 # obtain audio from the microphone
145 r = sr.Recognizer()
146 with sr.Microphone() as source:
147
148     print("Say something!")
149     audio = r.listen(source)
150
151
152 # recognize speech using Google Speech Recognition

```

```

File explorer  Variable explorer  Find in files  Help
IPython console
Console 1/A
...: cv2.imshow("grayscale_blurr",gray)
...: cv2.waitKey(0)
...: #
...: # write the grayscale image to disk as a temporary file so we can apply
OCR to it
...: #
...: filename = "{}.png".format(os.getpid())
...: print (filename)
...: cv2.imwrite(filename, gray)
3453.png
Out[4]: True
In [5]: text = pytesseract.image_to_string(Image.open(filename))
...: print(text)
...: os.remove(filename)
...:
...: text=text.split()
...: test1=' '.join(text)
I am happy to see my mother today.
She is a wonderful mother ,I m so blessed to haver her as my mother.

```

STEP 4:

After having loaded the input required for the system to carry out operations . We next load the dataset that are required for training the neural network and neural network to predict the input. The given dataset is divided into dependent and independent data variables , X and Y.

Index	content	sentiment
0	@tiffanylue i know i was listenin to bad habit earlier and i started freakin at his part =[empty
1	Layin n bed with a headache ughhhh...waitin on your call...	sadness
2	Funeral ceremony...gloomy friday...	sadness
3	wants to hang out with friends SOON!	enthusiasm
4	@dannycastle We want to trade with someone who has Houston tickets, but no one will.	neutral
5	Re-pinging @ghostridah14: why didn't you go to prom? BC my bf didn't like my friends	worry
6	I should be sleep, but im not! thinking about an old friend who I want. but he's married now. damn, & he wants me 2! scandalous!	sadness
7	Hmmm. http://www.djhero.com/ is down	worry
8	@charviray Charlene my love. I miss you	sadness
9	@kelcouch I'm sorry at least it's Friday?	sadness
10	cant fall asleep	neutral
11	Choked on her retainers	worry
12	Ugh! I have to beat this stupid song to get to the next rude!	sadness
13	@BrodyJenner if u watch the hills in london u will realise what tourture it is because were weeks and weeks late i just watch itonlinelol	sadness
14	Got the news	surprise
15	The storm is here and the electricity is gone	sadness
16	@annarosekerr agreed	love
17	So sleepy again and it's not even that late. I fail once again.	sadness
18	@PerezHilton lady gaga tweeted about not being impressed by her video leaking just so you know	worry
19	How are YOU convinced that I have always wanted you? What signals did I give off...damn I think I just lost another friend	sadness
20	@raaaaaaek oh too bad! I hope it gets better. I've been having sleep issues lately too	worry
21	Wondering why I'm awake at 7am, writing a new song, plotting my evil secret plots muahahaha...oh damn it, not secret anymore	fun
22	No Topic Maps talks at the Balisage Markup Conference 2009 Program online at http://tr.im/mL6Z (via @bobdc) #topicmaps	neutral
23	I ate Something I don't know what it is... Why do I keep Telling things about food	worry
24	so tired and i think i'm definitely going to get an ear infection. going to bed "early" for once.	sadness

Format

Resize

☒ Background color

☒ Column min/max

Cancel

OK

x - Series

Index	content
0	@tiffanylue i know i was listenin to bad habit earlier and i started freakin at his part =[
1	Layin n bed with a headache ughhhh...waitin on your call...
2	Funeral ceremony...gloomy friday...
3	wants to hang out with friends SOON!
4	@dannycastillo We want to trade with someone who has Houston tickets, but no one will.
5	Re-pinging @ghostidah14: why didn't you go to prom? BC my bf didn't like my friends
6	I should be sleep, but im not! thinking about an old friend who I want. but he's married now. damn, & he wants me 2! ...
7	Hmmm. http://www.djhero.com/ is down
8	@charviray Charlene my love. I miss you

Format

Resize

☒ Background color☒ Column min/max

Cancel

OK

y - Series

Index	sentiment
0	empty
1	sadness
2	sadness
3	enthusiasm
4	neutral
5	worry
6	sadness
7	worry
8	sadness

Format

Resize

☒ Background color☒ Column min/max

Cancel

OK

STEP 5:

Any machine learning model accepts and process input in the machine learning format 0's and 1's. If the input given was is any other format other than 0's and 1's several preprocessing has to be done to ,convert into acceptable format.

In our system ,the data set what we have given is in string's of character the independent variable is converted by using method called 'one hot encoder' and independent variable is converted by using NLP by using Counter Vectorizer ,which following preprocessing (text cleaning) are carried out before.

```
# =====
for i in range(0,len(dataset)):
    #Removing any special character and digits
    test_train=re.sub('[^a-zA-z]', ' ',dataset['content'][i])
    #Converting into lower case
    test_train=test_train.lower()
    #Making each data point as list words
    test_train=test_train.split()
    # Removing stopwords from train data
    test_train=[ps.stem(word) for word in test_train if not word in set(stopwords.words('english')) ]
    test_train=' '.join(test_train)
    corpus.append(test_train)

# =====
# Preparing feature vector
# =====
#Convert a collection of text documents to a matrix of token counts

max_features = 3000
cv=CountVectorizer(max_features=max_features)
x=cv.fit_transform(corpus).toarray()
```

x - NumPy array

	187	188	189	190	191	192	193
1207	0	0	0	0	0	0	0
1208	0	0	0	0	0	0	0
1209	0	0	0	0	0	0	0
1210	0	0	0	0	0	0	0
1211	0	0	0	1	0	0	0
1212	0	0	0	1	0	0	0
1213	0	0	0	0	0	0	0
1214	0	0	0	0	0	0	0
1215	0	1	0	0	0	0	0
1216	0	0	0	0	0	0	0

Format

Resize

☒ Background color

STEP 6:

The next step is to train the neural network with the preprocessed data. The choice of number of hidden layers, input nodes and output nodes are of arbitrary decision. For this model the system works well with one hidden layer, input nodes set to 3000 and output nodes set to 6000. And the neural network is set to train.

```
252 '''
253
254 Building Artificial Neural Networks
255
256 '''
257 #Creating neural network layer object
258 classifier = Sequential()
259
260 #Adding first layer (the input layer) of the ANN
261 classifier.add(Dense(output_dim = 6000, init = 'uniform', activation = 'relu', input_dim=3000 ))
262
263 #Drop certain nodes in the network to avoid overfitting
264 classifier.add(Dropout(rate = 0.1))
265
266 #Adding first hidden layer
267 classifier.add(Dense(output_dim = 3000, init = 'uniform', activation = 'relu'))
268
269 #Drop certain nodes in the network to avoid overfitting
270 classifier.add(Dropout(rate = 0.1))
271
272 #Output layer
273 classifier.add(Dense(output_dim = 12, init = 'uniform', activation = 'sigmoid'))
274
275 #Compiling the network
276 classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
277
278 #Fitting the ANN model with preprocessed training and test data set
279 classifier.fit(x, y_cat, batch_size = 10, nb_epoch = 50)
280
281 #
282 # Looping over the dataset for each sentence to perform text processing steps for each datapoint
283 #
284 combined_all=[0,0,0,0,0,0,0,0,0,0,0,0]
285
286 for m in range (0,number_of_sentences):
287     corpus=[]
288
289     for i in range (0,len(dataset)):
290         #Removing any special character and digits
291         test_train=re.sub('[^a-zA-z]', ' ',dataset['content'][i])
292         #Converting into lower case
293         test_train=test_train.lower()
```

test_train	str	1	mopedronin bullet train tokyo ...
text	list	23	['I', 'am', 'happy', 'to', 'se...
x	int64	(40000, 3000)	ndarray object of numpy module
x_test	list	2	['I am happy to see my mother ...
y	Series	(40000,)	Series object of pandas.core.series module
y_cat	float64	(40000, 12)	[[0. 1. 0. ... 0. 0. 0.] [0. 0. 0. ... 1. 0. 0.]
y_cat1	int64	(40000,)	[2 10 10 ... 7 5 7]

File explorer

Variable explorer

Find in files

Help

IPython console

Console 1/A

```
...:
...: #Output layer
...: classifier.add(Dense(output_dim = 12, init = 'uniform', activation =
'sigmoid'))
...:
...: #Compiling the network
...: classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy',
metrics = ['accuracy'])
...:
...: #Fitting the ANN model with preprocessed training and test data set
...: classifier.fit(x, y_cat, batch_size = 10, nb_epoch = 50)
__main__:4: UserWarning: Update your 'Dense' call to the Keras 2 API:
`Dense(activation='relu', input_dim=3000, units=6000,
kernel_initializer='uniform')`
__main__:10: UserWarning: Update your 'Dense' call to the Keras 2 API:
`Dense(activation='relu', units=3000, kernel_initializer='uniform')`
__main__:16: UserWarning: Update your 'Dense' call to the Keras 2 API:
`Dense(activation='sigmoid', units=12, kernel_initializer='uniform')`
/Users/apple/anaconda3/lib/python3.6/site-packages/keras/models.py:981:
UserWarning: The 'nb_epoch' argument in 'fit' has been renamed 'epochs'.
warnings.warn('The `nb_epoch` argument in `fit` '
Epoch 1/50
1820/40000 [>.....] - ETA: 22:56 - loss: 0.2748 - acc:
0.9104
```

STEP 7:

After training the neural network. The input sentence/characters obtained from earlier by loading the input scanned image to ocr model is used to predict the sentence emotions into 12 classes. The output obtained at the is given in the figure.

I am happy to see my mother today.

Boredom = 0.02 %
Empty = 0.9 %
Enthusiam = 1.61 %
Fun = 3.59 %
Happiness = 30.22 %
Hate = 0.16 %
Love = 36.8 %
Netural = 18.63 %
Relief = 4.6 %
Sadness = 2.85 %
Suprise = 5.62 %
Worry = 8.48 %

She is a wonderful mother, I am so blessed to have her as my mother.

Boredom = 0.0 %
Empty = 0.11 %
Enthusiam = 0.39 %
Fun = 1.72 %
Happiness = 19.07 %
Hate = 0.07 %
Love = 58.22 %
Netural = 10.06 %
Relief = 1.66 %
Sadness = 0.89 %
Suprise = 9.17 %
Worry = 8.08 %