Detecting Emotion in Text and Speech

NLP with Watson Final Project Paper

Sarabeth Jaffe

1.1 Summary

The goal of my final project was to compare the effectiveness in determining emotions through speech (using indicators such as tone, pace and volume) with determining the emotion underlying text (using keyword scoring, sentiment analysis, punctuation examination and much more.) As of now, I have a working prototype of a Java-based system that hypothesizes the emotional label of a given audio file and its text transcription.

1.2 Goals

My goal for this project was to determine the importance of the components of speech versus text translation in terms of detecting emotion. The original questions I hoped to answer by the completion of my project are the following which I will provide my findings for.

1. *Is it more effective to solely analyze speech, solely analyze text, or create a scoring system to combine the two?*

   In my case and with the techniques I used, it was much more effective to analyze speech. However, with more advanced text-processing and machine learning techniques, it would most likely be much more effective to weigh the speech score with the advanced text score.

2. *How important are certain indicators in speech for specific emotions?*

   In my research, I found that the most revealing indicators for speech included: volume, silence regions and pace (words per second.) Anger was the loudest and had the most average words per second (2.08/second) and sadness had the longest silent regions.

3. *How should I weigh the speech score with the textual score?*

   With further implementation on the text side, I believe they should be scored together in the future. The most important indicators that should have large scoring weight include: volume, silent regions and pace along with sentiment scoring, adjective detection and positive and negative phrase detection.

4. *Even if you use "positive" keywords, can your speech patterns reveal your true emotions?*

   This question is still unknown as it would rely on the scoring techniques used and would also possibly require a voice actor or a very specific audio clip.

Overall, my main goal and challenge for this project was to learn as much as I could, especially in speech analysis and I believe I accomplished that.

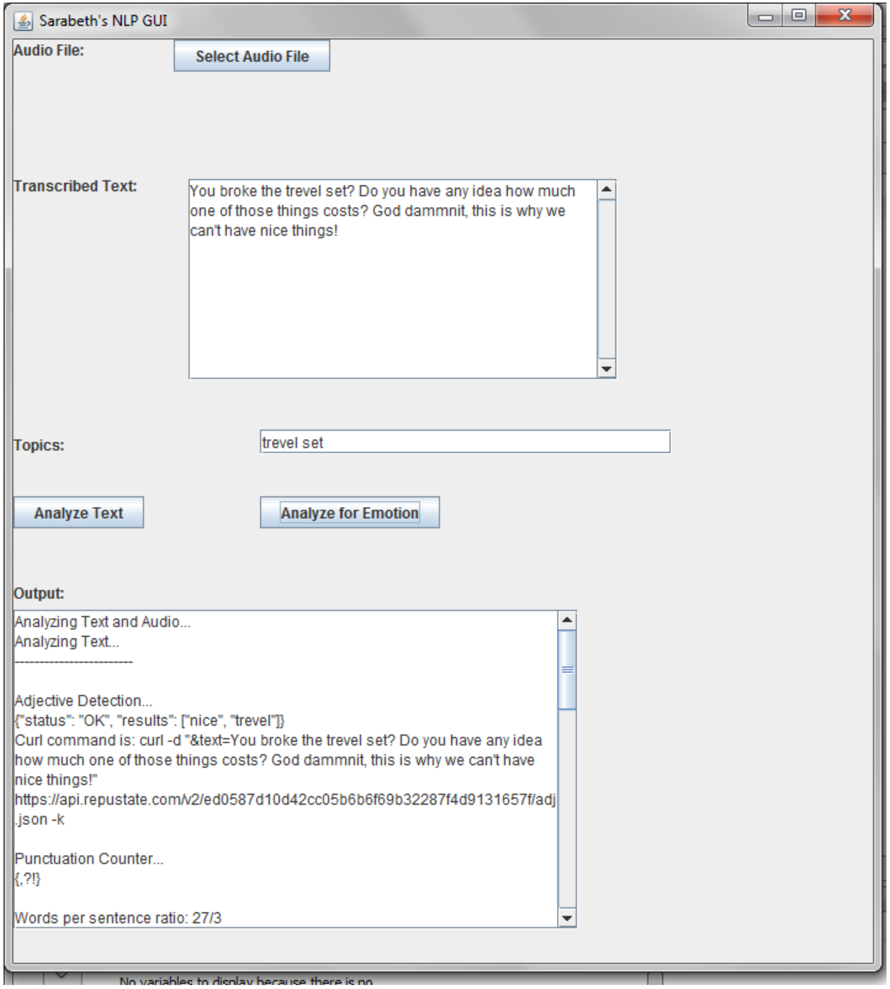
1.3 Deliverables

Right now my project is a working prototype that focuses on detecting three simple emotions: Joy, Anger and Sadness. I chose these emotions because Joy can easily be filtered out from Anger and Sadness just based on a sentiment scoring system.

On the text side, I have incorporated sentiment scoring (positive, negative, neutral,) topic-based sentiment scoring, adjective detection, keyword detection, and punctuation detection.

On the speech side, I have done a lot of audio analysis by hand and through the use of different software. For example, see figures 1, 2 and 3 to view the great differences in the decibel frequency shown in the different emotions. These frequency analysis graphs are plotted by dB[1] (decibels) versus Hz[2] (Hertz) using the "Plot Spectrum" functionality in Audacity[3]. These plots allow you to easily differentiate emotions by their amplitude. Figure 4 demonstrates further functionality of Audacity.

Below is the interface and example output/input to the system.



Transcribed Text: You broke the trevel set? Do you have any idea how much one of those things costs? God dammnit, this is why we can't have nice things!

Topics: trevel set

Output:

```
Analyzing Text and Audio...
Analyzing Text...
------------------------


Adjective Detection...
{"status": "OK", "results": ["nice", "trevel"]}
Curl command is: curl -d "&text=You broke the trevel set? Do you have any idea
how much one of those things costs? God dammnit, this is why we can't have
nice things!"
https://api.repustate.com/v2/ed0587d10d42cc05b6b6f69b32287f4d9131657f/adj
.json -k

Punctuation Counter...
{,?!}

Words per sentence ratio: 27/3
```

[1] A logarithmic unit (typically of sound pressure) describing the ratio of that unit to a reference level.
[2] Measures a frequency event in number of cycles per second.
[3] Audacity is free, open source, cross-platform software for recording and editing sounds.

Total output from the system includes: adjective detection, punctuation counter/detector, words per sentence ratio, keyword detection, sentiment scoring and topical emotions analysis. For speech, output includes overall loudness, word tempo (pace,) periods of silence, overall pitch, words per second and an emotion prediction.

1.4 Resources Used

More resources used include Moodzle API[4], a system that is capable of speech emotion recognition that claims to hold a 91% accuracy level on untrained data. Unfortunately, part way through my project their servers went down and I'm unable to use their services as of now. I also used Vamp[5], an audio analysis plugin used for audio feature extraction. Lastly, I used Sonic Visualizer[6], a program that allows you to view audio graphs in meaningful ways.

I also used the sentiment dictionary from Columbia University for the detection of positive, negative and negation indicators.

For the main sample audio clips I used snippets of a video from YouTube entitled, "30 Emotions Voice Acting Exercise."

1.5 Improvements

There are many ways that I would like to improve my system. For the text side, I would like to implement curse word detection. It would be interesting to test which curse words are used the most in which emotion. (Of course this could vary a lot on a person by person basis.) The reason I think this would be an interesting problem to address is because in my anger audio there is some slight cursing which brings its sentiment score very low. However, for my happiness audio, there is the expression "Oh my god" which seems to be the reason its sentiment score comes back as "neutral" even though it also contains very positive words.

I would also like to either get my hands on a list of emotive expressions or implement the detection of emotional expressions. I came to find that phrase detection would actually be more useful than keyword detection when extracting emotions because you need more context as to what's going on with the speaker. I think I should have actually taken a route more similar to Event Extraction in that it's important to get a deeper sense of the meaning of the text when dealing with emotions. The more information we can give to the system, the better because, while it's hard to get a computer to understand language, it's a whole other story trying to get it to understand human emotion.

Another step to take would be to analyze a large corpus of speech and text files in order to prove/improve my hypotheses.

[4] "EmotionAPI is capable of listening to the human voice and determining which emotions are present in the voice, as well as transcribing the speech into text. [Moodzle API] generate[s] emotion labels by studying scores and scores of measurements which can be applied to the speech signal, and then classifying the most likely emotion based on all the measurements."

[5] Vamp is an audio processing plugin system that extracts descriptive information from audio data — typically referred to as audio analysis plugin or audio feature extraction plugin.

[6] Sonic Visualiser is an application for viewing and analysing the contents of music audio files.

I also would have like to get automatic audio to transcription implementation down either through Moodzle or use another API such as MyCaption.[7]

Lastly, I would have liked to get automatic extraction of audio features done using Sonic Annotator. Sonic Annotator would allow me to extract audio features and output select information to a file using RDF formatting.

1.6 Relatedness to Watson

IBM has made the term "cognitive computing" big after their reveal of Watson, their Jeopardy playing supercomputer. But just how cognitive is Watson? IBM's definition of cognitive systems is the following: "a category of technologies that uses natural language processing and machine learning to enable people and machines to interact more naturally to extend and magnify human expertise and cognition."[8] A system that would be able to detect emotion from the user would greatly help "people and machines interact more naturally." Adding an emotion detection component to an information retrieval system like Watson would allow a system to both "think" and "feel," connecting the fields of cognitive computing and affective computing.

1.7 Conclusions

Overall, I learned a lot about audio analysis, using and utilizing APIs, and how to think of creative ways of incorporating different NLP techniques into one system.

However, one of my biggest frustrations throughout the project was spending so much time trying to set-up different systems and tools which ended up not functioning at all or as expected.

1.8 Future Research

While researching for this project I came along a paper discussing keystroke dynamics. I would be interested in answering questions such as: How do Keystroke Dynamics relate to the text typed and would they help improve emotion detection in text? If so, would there be a noticeable difference for different mediums such as typing on a keyword versus a touch-screen phone? How many typos or incorrect auto-corrects would come about for each emotion? Exactly how revealing is punctuation for less formal text-based communications?

References:

"Identifying Expressions of Emotion in Text," Saima Aman and Stan Szpakowicz.

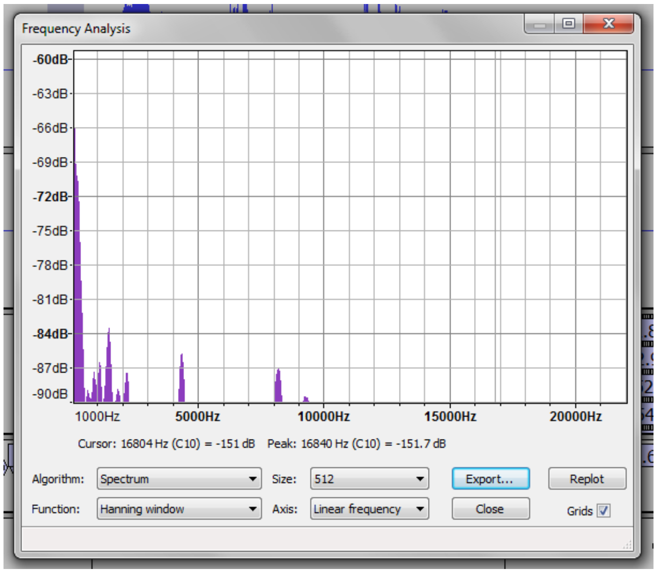
"Identifying Emotional Expressions, Intensities and Sentence level Emotion Tags using a Supervised Framework," Dipankar Das and Sivaji Bandyopadhyay.

"Emotion Detection from Text", Shiv Naresh Shivharel and Prof. Saritha Khethawat.

[7] I wasn't able to implement this because I need to set-up my own web server in order to receive the response.
[8] "Why cognitive systems?" IBM Research

Diagrams:

Figure 1: Angry Frequency Analysis

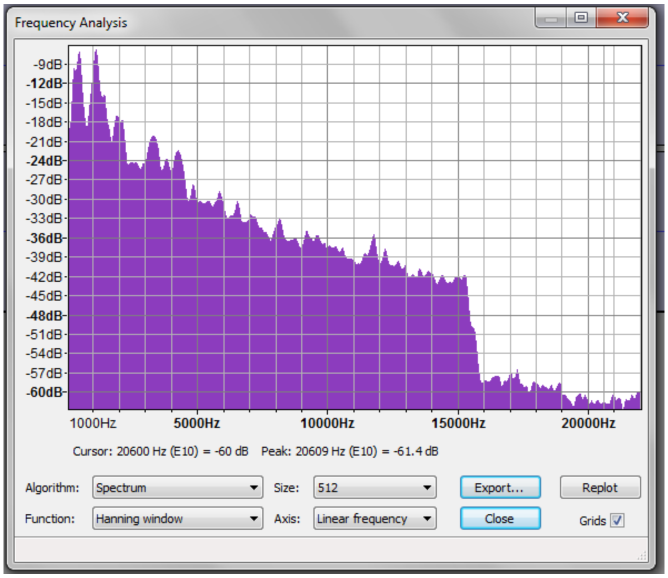


Cursor: 16804 Hz (C10) = -151 dB Peak: 16840 Hz (C10) = -151.7 dB

Algorithm: Spectrum Size: 512

Function: Hanning window Axis: Linear frequency Grids ☑

Figure 2: Happy Frequency Analysis

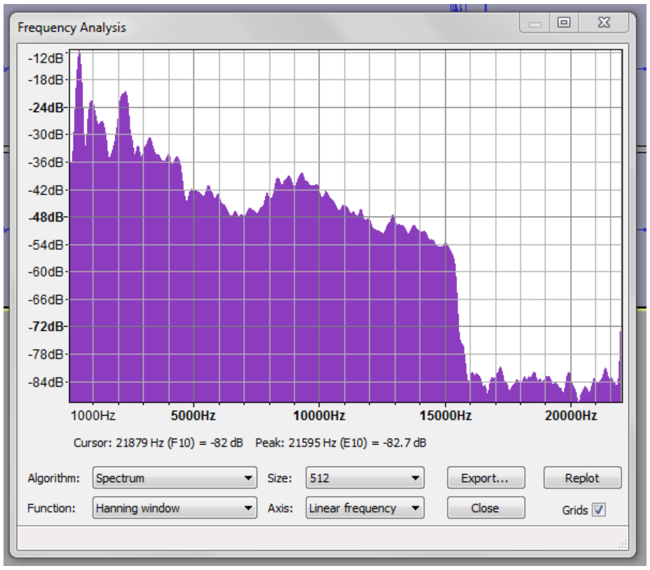


Cursor: 20600 Hz (E10) = -60 dB Peak: 20609 Hz (E10) = -61.4 dB

Algorithm: Spectrum Size: 512

Function: Hanning window Axis: Linear frequency Grids ☑

Figure 3: Sad Frequency Analysis:

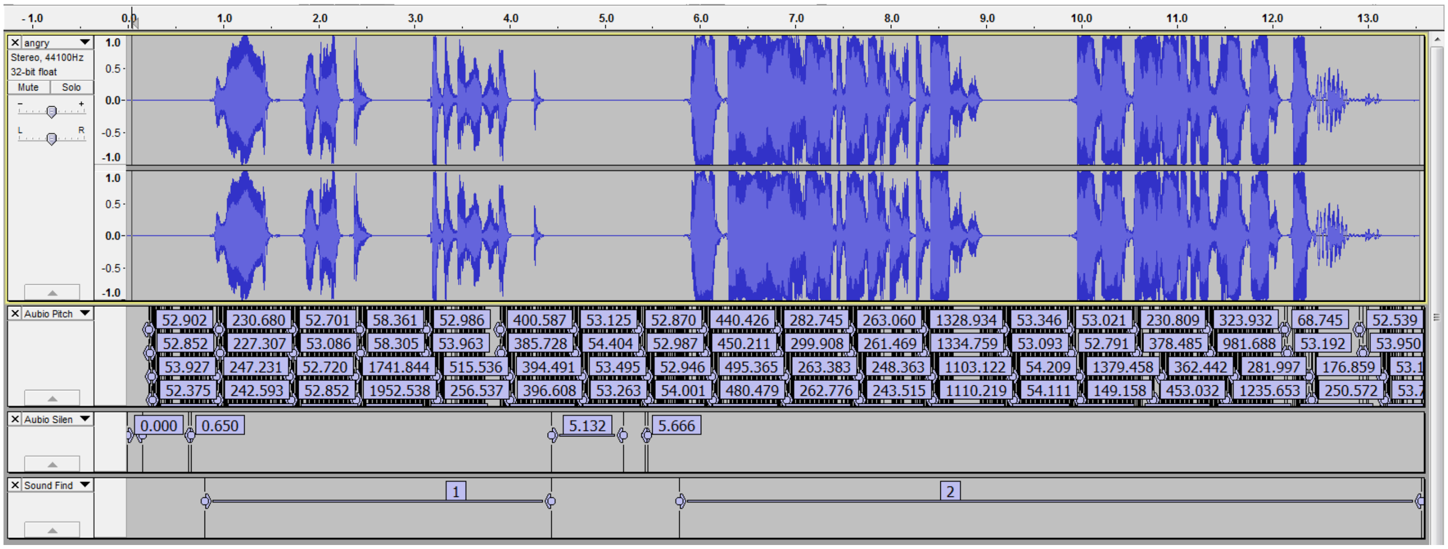


Cursor: 21879 Hz (F10) = -82 dB Peak: 21595 Hz (E10) = -82.7 dB

Algorithm: Spectrum Size: 512

Function: Hanning window Axis: Linear frequency Grids ☑

Figure 4: First row is the audio itself, second is pitch, third is silence detection and fourth is sound detection.


-1.0
0.0
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
9.0
10.0
11.0
12.0
13.0
angry
Stereo, 44100Hz
32-bit float
Mute
Solo
1.0
0.5
0.0
-0.5
-1.0
Aubio Pitch
52.902
52.852
53.927
52.375
230.680
227.307
247.231
242.593
52.701
53.086
52.720
52.852
58.361
58.305
1741.844
1952.538
52.986
53.963
515.536
256.537
400.587
385.728
394.491
396.608
53.125
54.404
53.495
53.263
52.870
52.987
52.946
54.001
440.426
450.211
495.365
480.479
282.745
299.908
263.383
262.776
263.060
261.469
248.363
243.515
1328.934
1334.759
1103.122
1110.219
53.346
53.093
54.209
54.111
53.021
52.791
1379.458
149.158
230.809
378.485
362.442
453.032
323.932
981.688
281.997
1235.653
68.745
53.192
176.859
250.572
52.539
53.950
53.1
53.7
Aubio Silen
0.000
0.650
5.132
5.666
Sound Find
1
2