**Review of Literature**

**Labeling emotions**

A user begins by selecting a speech track

such as an audiobook, and one or more instrumental music

tracks (i.e., music without lyrics). Our system then gathers

emotion labels for each of the speech and music tracks. After

collecting these labels, our system constructs a musical score

by re-sequencing the input music tracks so the emotion labels

of the output music match the emotion labels of the speech.

Our system offers three methods for obtaining the required

emotion labels: hand-labeling, crowd-labeling, and automatic

labeling. These three methods offer trade-offs in time, effort,

and personalization. Hand-labeling produces emotions that

reflect the user’s emotions and may result in the most personal

musical score, but it takes the most time for the user. The

crowd-labeling method requires no extra work on the user’s

part and incorporates human ratings. However, this method

takes more time than hand-labeling to acquire emotion labels.

Finally, the fully automatic method produces labels immedi-

ately, but they may not accurately reflect the personal emo-

tions that the user—or any human—feels about the story and

the music.

As we will show, all of these methods require

a speech transcript, which may cost money to obtain. The

crowd-labeling method has the added cost of paying workers.

Instead, we focus

on four emotions, happy, nervous, sad, and calm, because

they almost evenly span the circumplex

**Speech Emotion Labels**

Our goal is to break the speech into segments of similar emo-

tions.

Therefore, we obtain a text tran-

script of the speech and use its paragraph boundaries to seg-

ment the speech.

We time-align the text tran-

script and the speech using a variant of the Penn Phonetics

Lab Forced Aligner [36] from Rubin et al [22]. The time-

alignment allows our system to find the segment of speech

that corresponds to each paragraph of the text.

**Hand-labeling** the speech. If users wish to personalize the

emotion labels of the speech, they can label the emotion of

each paragraph of the text transcript by hand (Figure 3).

**Crowd-labeling** the speech. To crowd-label the emotions of

the speech, our system posts tasks to Amazon’s Mechanical

Turk. These tasks are identical to the hand-labeling interface

**Automatically labeling** the speech. To automatically label

the speech, our system estimates the emotion of each para-

graph based on the emotion conveyed by each word.

Warriner

et al. [31] have collected a corpus of crowdsourced valence

and arousal ratings for nearly 14,000 English words. We nor-

malize these scores by the global valence/arousal mean and

standard deviation. We then compute the average of the nor-

malized valence/arousal scores of all words in a paragraph.

Our system projects those averages to the nearest of our four

labels to obtain an emotion label for each paragraph.

**Music Emotion Labels**

Structural segments

of music often contain one predominant emotion because the

features that differentiate structure—timbre, pitch, volume,

and self-similarity—are also indicative of music emotion.

Following McFee and Ellis [15], our system segments mu-

sic by computing a hierarchical clustering of self-similarities

in a track and finding an optimal pruning of the cluster tree.

The user selects one of the three labeling methods to get emo-

tion labels for each music segment.

Hand-labeling the music. If users wish to personalize the

emotion labels of the speech, they can listen to and assign

labels for each music segment (Figure 3) As in hand-labeling

the speech, this method is preferable for users that have time

and desire a personalized musical score.

Crowd-labeling the music. To crowd-label the emotions of

the music, our system asks workers on Mechanical Turk to

listen to and label the emotions of the music segments for

an entire track (Figure 3). Our system then selects a final

emotion labeling by finding the worker’s labeling that best

represents all of the worker labelings (see earlier section on

Crowd-labeling the speech).

Automatically labeling the music. Schmidt et al. [24, 25]

have developed methods for automatically predicting the va-

lence and arousal of music. Their MoodSwings Turk dataset

consists of a large set of crowd-generated, per-second va-

lence/arousal labels and accompanying audio signal pro-

cessing features (MFCCs [14], spectral contrast [10], and

chroma [7]). Our goal is to automatically predict the emotion

of each music segment, but the dataset contains per-second

labels and no notion of segments. We follow Schmidt et

al. [24] and train a multiple linear regression model on the

MoodSwings Turk dataset to predict per-second emotions.

**Score generation algo**

To generate an emotionally relevant score, our

algorithm translates emotion matching and our other con-

straints into numeric costs. The two primary costs are match-

ing costs—costs of how well the music emotions match the

speech emotions—and transition costs—costs for beat-to-

beat transitions in the music. Once our algorithm computes

these costs, it searches for the lowest cost musical score using

dynamic programming.

We set m to the duration of the

speech track divided by the average beat length of the mu-

sic. Our algorithm searches for a sequence s of m beats that

minimizes the matching and transition costs; it then generates

the output musical score by re-sequencing the original beats

according to s.

**SENTIMENT ANALY**

**Text based sentiment analysis**

The process encompasses two phases:

(i) Pre-processing REFER PG 3

(ii) Attitude prediction REFER PG 3

Naive bayes or SVM training and testing

**audio sentiment analysis**

Audio features like pitch, intensity and loudness are extracted using Open- EAR software and Support Vector

Machine (SVM) classifier is built to detect the sentiment . The audio features are automatically extracted from each

video clip using OpenEAR software and Hidden Markov Models (HMM) classifier is built to detect the sentiment

1. **Preprocessing**

We have to divide speech signal into frames. Then we compare

each frame with phonemes label in the database and find the frame have a silence phoneme label and remove that

frame. After that, we merge whole speech frames into utterances again. Silence is considered as an useless data in this

research. After getting speech data, all of them are divided into frames. Each frame will be extracted features in below.

1. **Acoustic feature extraction**

All experiments in this paper are conducted only with utterance based features on training and testing the classifiers.

All of utterance-based features are concatenated together, before calculating the first and second derivatives of them.

Hidden Markov Model (HMM):-

Viterbi algorithm.