MUSIC BASED MOOD DETECTION

A Thesis Submitted to the Academic Faculty in Partial Fulfillment of the Requirements for the Degree of

BACHELOR OF SCIENCE IN COMPUTER SCIENCE & ENGINEERING

Prepared by,

Shahid Ishraq (Student No-134408)
Zia Uddin Kamal (Student No-134430)

Supervised by,

Hasan Mahmud
Assistant Professor
Department of Computer Science and Engineering,
Islamic University of Technology (IUT)

A thesis submitted to the Department of CSE
in partial fulfillment of the requirements for the degree of B.Sc.
Engineering in CSE
Academic Year: 2016-17
November – 2017

Declaration of Authorship

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by Shahid Ishraq and Zia Uddin Kamal under the supervision of Hasan Mahmud, Assistant Professor, Department of Computer Science and Engineering (CSE), Islamic University of Technology (IUT), Dhaka, Bangladesh. It is also declared that neither of this thesis nor any part of this thesis has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others has been acknowledged in the text and a list of references is given.

Authors:

____

Shahid Ishraq

Student ID – 134408

____

Zia Uddin Kamal

Student ID – 134430

Approved By:

____

Hasan Mahmud
Supervisor and Assistant Professor

Department of Computer Science and Engineering,

Islamic University of Technology (IUT)

Acknowledgement

We would like to express our grateful appreciation for Hasan Mahmud, Assistant Professor, Department of Computer Science & Engineering, IUT for being our advisor and mentor. His motivation, suggestions and insights for this thesis have been invaluable. Without his support and proper guidance this research would not have been possible. His valuable opinion, time and input provided throughout the thesis work, from first phase of thesis topics introduction, subject selection, proposing algorithm, modification till the project implementation and finalization which helped us to do our thesis work in proper way. We are really grateful to him. We are also grateful to Dr. Kamrul Hasan, Associate Professor, Department of Computer Science & Engineering, Islamic University of Technology for his valuable suggestion.

Abstract:

Music mood describes the inherent emotional expression of a music clip. It is helpful in music understanding, music retrieval, and some other music-related applications. In this paper, a hierarchical framework is presented to automate the task of mood detection from acoustic music data, by following some music psychological theories in western cultures. The hierarchical framework has the advantage of emphasizing the most suitable features in different detection tasks. Three feature sets, including intensity, timbre, and rhythm are extracted to represent the characteristics of a music clip. The intensity feature set is represented by the energy in each sub-band, the timbre feature set is composed of the spectral shape features and spectral contrast features, and the rhythm feature set indicates three aspects that are closely related with an individual's mood response, including rhythm strength, rhythm regularity, and tempo. Furthermore, since mood is usually changeable in an entire piece of classical music, the approach to mood detection is extended to mood tracking for a music piece, by dividing the music into several independent segments, each of which contains a homogeneous emotional expression. Preliminary evaluations indicate that the proposed algorithms produce satisfactory results. On our testing database composed of 800 representative music clips, the average accuracy of mood detection achieves up to 86.3%. We can also on average recall 84.1% of the mood boundaries from nine testing music pieces.

A method is proposed for detecting the emotions of song lyrics based on an affective lexicon. The lexicon is composed of words translated from ANEW and words selected by other means. For each lyric sentence, emotion units, each based on an emotion word in the lexicon, are found out, and the influences of modifiers and tenses on emotion units are taken into consideration. The emotion of a sentence is calculated from its emotion units. Tofigure out the prominent emotions of a lyric, a fuzzy clustering method is used to group the lyric's sentences according to their emotions. The emotion of a cluster is worked out from that of its sentences considering the individual weight of each sentence. Clusters are weighted according to the weights and confidences of their sentences and singing speeds of sentences are considered as the adjustment of the weights of clusters. Finally, the emotion of the cluster with the highest weight is selected from the prominent emotions as the main emotion of the lyric. The performance of our approach is evaluated through an experimentof emotion classification of 400 song lyrics.

Therefore, the main idea is to merge the two ideas altogether.

CONTENTS

1. Introduction

"Music heals everything." - Hermeto Pascoal

Nowadays, music is everywhere. From small television commercials, music videos and shopping centers to the more traditional music sell (albums, both on physical or digital format, and tickets), radio play or live events (concerts, gigs, festivals, etc), we cannot help consider now it as an industry. This can be explained through economical reasons but also through emotional ones: why some concerts get sold out in just a few minutes1, even when the tickets price is high?1 Why are there true fan communities online which create means to bring some of their most beloved artists to their countries or cities (petitions, organizing gigs/concerts, etc)2 ? And why big part of humanitarian and charity events are based in (or incorporate) music (e.g., Live 8)?3 Certainly not only because of temporary fashions or trends. Even some of the most popular social networks on the web are music-oriented: MySpace4 (with more than 110 million active users5) and Last.fm6 (with more than 30 million users7). People need to relate to the way they feel (or want to feel), and music is one of the most utilized means to achieve it. This is done not only by relating to the instrumental part of the songs, but in the case of the existence of voice, the analysis of the lyrical content.

1.1. The Context

These are only some points that relate music to mood and emotions. Later it will be defined and discussed in what the concepts of emotion and mood resemble, differ and relate to each other.
In this context, it is easy to understand that mood analysis has gained increased notoriety in the last few years, with increasing popularity coming from research in the Music Information Retrieval (MIR) field in the last decade (where the Music Emotion Retrieval (MER) area emerged).
was the inclusion, since 2007, of a Music Mood Classification evaluation contest in the 3rd Music Information Retrieval Evaluation eXchange (MIREX)8, a part of the 8th International Conference on Music Information Retrieval (ISMIR) (ISMIR 2007)9 . ISMIR is the most important conference dedicated to MIR in the world.

Possible MIR applications include automatic music recognition, automatic cataloging of musical pieces and automatic generation of music playlists based in a similarity criterion (mood, style, etc). The term "automatic" is used explicitly to underline the fact that these playlists and cataloging are produced based entirely on the extraction and analysis of songs features, without the use of any kind of descriptive text tags (typical ones are artist, album, genre, etc.).

In the more specific context of this thesis (mood analysis), the applications can be extended to the generation of music playlists based in certain moods, without bothering the user with the task of browsing his personal musical collection (which can easily reach (dozens of) thousands of songs) to manually choose the songs. This will simultaneously be both more comfortable and time sparing to the user. For instance, if the user wants music to play while he is jogging or doing sports (imagining the application is being used in a portable music player) he may want to hear some joyful, "happy" songs, beat-driven and/or with rhythm. On the other hand, if someone is driving he may want to be presented with some quiet, relaxing music. The same would apply to a stressful situation, where the user just wants to sit and calm down, or a psychological treatment. Even a shop owner or a DJ could benefit from this type of application, by automatically selecting cheerful music to serve as his store's background music or selecting music based on the mood of the venue he is playing in (excitement for a club where rock tunes are played traditionally, for instance), respectively. Finally, another possible daily situation where this type of software could be applied in a useful way is the selection of music for a party, where the user could choose the input mood based on the theme (or even dress code) of it.

1.2. The Problem

Although at first glance mood and emotion can be easily confused, the two concepts are considered to be different. As a subjective theme, everyone knows what an emotion is, but rarely one can define it. Since then, psychologists have made several attempts to achieve definitions of emotion and mood and a reliable model of human emotion in the mind.

In the BSc thesis "A Mood-Based Music Classification and Exploration System" by Meyers (Meyers, 2007), emotion is defined as:

> "Emotion is a complex set of interactions among subjective and objective factors, mediated by neural/hormonal systems, which can (a) give rise to affective experiences such as feelings of arousal, pleasure/displeasure; (b) generate cognitive processes such as perceptually relevant effects, appraisals, labeling processes; (c) activate widespread physiological adjustments to the arousing conditions; and (d) lead to behavior that is often, but not always, expressive, goal-oriented, and adaptive."

In the same thesis, mood is interpreted as something more specific, shorter emotional state and that can be (indirectly) influenced by the surroundings, the "environment" around the individual, so to speak. It can be attributed to a particular stimulus and usually has a prolonged effect.

1.3. The Solution

The main objective of this thesis is, as the title suggests, the design and implementation of an application that provides the user with an automatically generated playlist of songs based on the mood of a song specified by the user and given as input. In another perspective, it can be used to generate a playlist to complement the mood of the user – e.g., the user introduces that his mood is anxious, it will be expected that the playlist will include some relaxing music. The song given as input will be an existing song in the application's database (through a query). The selection would be done based on a database consisting of several music files/musical pieces. The pieces that would fit the mood specified by the user (or the pieces that would complement the specified mood) would be added to the final playlist presented to it. The user would then be presented with several options to customize these operations. Details about how this will be done are referred to and discussed later in this document.

1.4. Innovative Aspects

The approach carried out on this thesis involves two aspects: it is both an engineering and a research problem. It is a typical engineering problem in what concerns both to the paradigm (a client-server application, with a backoffice for database/application management) and to the software development process used – in this case, the waterfall process, with the typical phases associated to it: requirements specification, design, implementation, model validation and testing/debugging. The investigation started with the bibliographic search that was made and that will be detailed later on this report. The research also focused on the choice of the framework to be used to extract features from songs. From the three frameworks analyzed (jAudio10, Music Analysis, Retrieval and Synthesis for Audio Signals (MARSYAS)11 and MIRtoolbox12, a toolbox for MATLAB), the one that proved to be faster was MARSYAS (performance results in the MIREX 2008 context prove it)13, since it is implemented in highly optimized C++ code. It is also important to underline that although both MARSYAS and jAudio are open source, the latter one, as the name suggests, is implemented in Java, which must certainly be the main factor for not being as fast as MARSYAS.

On the other hand, MIRtoolbox uses MATLAB, which is also partially implemented in Java, making it heavy in what concerns to computational requirements and also much slower than MARSYAS. Also, from the three frameworks compared, only MATLAB is not open source. So, MARSYAS was naturally chosen as the framework to be used within this project. Since it is open source, it is possible to add other functionalities to the application in the future (for instance, new features to extract, new classifiers, etc). The application also has a Graphical User Interface (GUI), which was implemented in Qt14 (a fast and open source application for creating interfaces). This decision was basically based on the fact that Qt is a cross-platform application and UI framework. Also, MARSYAS is natively prepared for seamless integration with Qt.

In this work, we follow a classification-based song similarity analysis. A classifier is first trained and then song similarity is calculated based on the distance between songs in Thayer's arousal-valence plane. Here, Support Vector Machines (SVM) are employed. Moreover, an extensive research of the features available in the three frameworks analyzed was conducted. This research is presented later in the document. As mentioned, the emotional model to be used is the Thayer model, which will also be discussed in detail later.

In what concerns to the GUI, the application presents the user with a graphical representation of the Thayer's model, where songs are represented as dots in the arousal / valence plane, and each of the four quadrants has an appropriated color (e.g. blue for depression and green for contentment, with a color gradient inside each quadrant). Also, for the creation of the playlist, a playlist path in the Thayer's plane is drawn by the user, from which songs are obtained.

The application that is the subject of this thesis was co-developed with a colleague, Renato Panda. The core of the application is similar, although Renato's thesis is focused on automatic mood tracking in audio music (mood change analysis during one song), opposed to this thesis theme (automatic playlist generation via music mood analysis). The main core of the application was developed in team work, while the functionalities related to playlist generation were the subject of this thesis and functionalities related to mood tracking were implemented by Renato.

A method is proposed for detecting the emotions of song lyrics based on an affective lexicon. The lexicon is composed of words translated from ANEW and words selected by other means. For each lyric sentence, emo-tion units, each based on an emotion word in the lexicon, are found out, and the influences of modifiers and tenses on emotion units are taken into consideration. The emo-tion of a sentence is calculated from its emotion units. To figure out the prominent emotions of a lyric, a fuzzy clus-tering method is used to group the lyric's sentences accord-ing to their emotions. The emotion of a cluster is worked out from that of its sentences considering the individual weight of each sentence. Clusters are weighted according to the weights and confidences of their sentences and singing speeds of sentences are considered as the adjust-ment of the weights of clusters. Finally, the emotion of the cluster with the highest weight is selected from the promi-nent emotions as the main emotion of the lyric. The perfor-mance of our approach is evaluated through an experiment of emotion classification of song lyrics.

1.5. Structure of the Thesis

AUDIO BASED APPROACH:

The planning and schedule of this thesis is now presented. It suffered some changes, basically due to the inclusion of new elements on the project and to the update on the application requirements (the project grew in dimension over time). It is important to notice that the time that was initially planned for the state of the art document revealed short, since the bibliographic research and reading proved to be more time consuming. Also, some delay was introduced due to software experimentation (more specifically MARSYAS) – although its open source nature, documentation is almost inexistent (some manuals exist, but they are still incomplete). So, on the final planning Gantt diagram some of the phases of the project became overlapped in some degree.

The second part (semester) consisted mainly in the application implementation, evaluation and testing. Also, some evaluation tests were performed, in what concerns to classification accuracy and playlist generation.

LYRIC BASED APPROACH:

In order to organize and search large song collections by emotions, we need automatic methods for detecting the emotions of songs. Especially, they should work in small devices such as iPod and PDA. At present, much, if not most, research work on song emotion detection was con-centrated on the audio signals of songs. For example, a number of algorithms [2,7,9] that classify songs from their acoustic properties were developed.

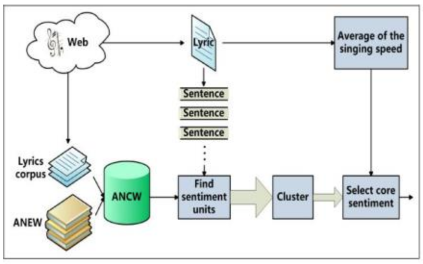
The lyric of a song, which will be heard and understood by listeners, plays an important part in determining the emotion of the song. Therefore, detecting the emotions of the lyric effectively contributes to detecting the emotions of the song. However, there is now comparatively less research done on methods for detecting the emotions of songs based on lyrics. There has been indeed a very large

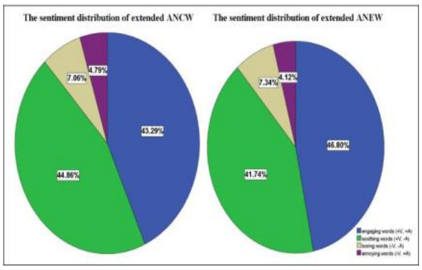
literature already out there on emotion analysis or opinion analysis of text. But, nearly all of them [1, 3, 6] use a one-dimensional model of emotions, such as positive-negative, which is not fine enough to represent lyric emotions which need more dimensions. Lyrics are much smaller in size than other kinds of text, such as Weblogs and reviews, and this makes it hard to detect lyrics' emotions. Being more challenging, lyrics are often abstract and in lyrics, emotions are expressed implicitly.

We propose an approach to detecting the emotions of lyrics based on an affective lexicon. The lexicon is orig-inated from a translated version of ANEW and then ex-tended. According to the lexicon, emotion units(EUs) [13] of a sentence are extracted and the emotion of the sentence is calculated from those EUs.

A lyric generally consists of several sentences and those sentences usually expresses more than one emotions. In order to figure out all the prominent emotions of a lyric, we use a fuzzy clustering method on the sentences of the lyric. The method is robust enough to sustain the noises induced in previous processing steps.

In our approach, Russell's model of mood [11] is adopted, as shown in Figure 1, in which emotions are represented by two dimensions, *valence* and *arousal*. The lyric files we use are in LRC format [1] which have time tags in them and we got the LRC files from the Web. The framework of our approach is illustrated in Figure 2. It consists of three





main steps: (i) building the affective lexicon (ANCW); (ii) detecting the emotion of a sentence; (iii) integrating the emotions of all sentences.

The rest of this paper is organized as follows. In Section 2, the method for building an affective lexicon is presented. Section 3 describes the method for detecting the emotions of sentences. The approach to integrating the emotions of sentences is described in Section 4. Experiments and dis-cussion are presented in Section 5. Finally, we conclude our work in Section 6.

2. <u>State of the Art</u>

2.1.1. Mood taxonomies

Among the several papers addressed in the last paragraphs, many distinct mood categories and taxonomies where proposed. One typical and common problem in this area is the existence of dozens (even hundreds or thousands) of different words or terms to describe moods, most them describing somewhat similar and redundant ones. Usually these words are adjectives, but there is the need of normalizing the terms used, since there is not a standard mood taxonomy. Mood taxonomies can be grouped in two main approaches: categorical and dimensional.

Thayer's model

A popular and more recent approach is based on the simple Robert Thayer's Model ((Meyers, 2007), (Lu, Liu, & Zhang, 2006), (Laar, 2005)), a dimensional model where a musical piece can be classified in one of four categories mapped into a four-quadrant, bi-dimensional space, as seen in Figure below:

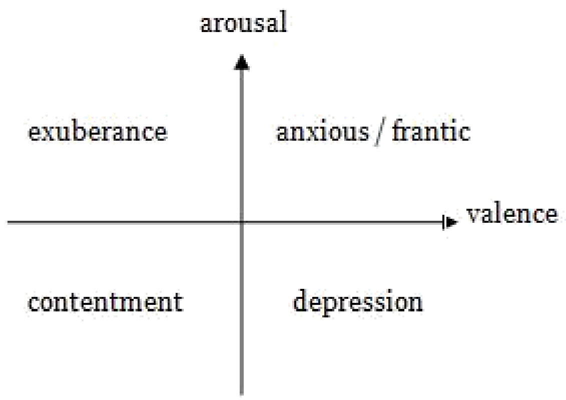


Figure 4: Thayer's dimensional model (Lu, Liu, & Zhang, 2006)

This approach has its advantages, since although simple it relates two different dimensions effectively, and so each category could be the result of some combination of the valence and arousal components. In the horizontal axis (valence) the quantity of stress is measured, while the vertical axis (arousal) denotes the quantity of energy. Although the model consists of four basic moods, the relation between the two amounts (coordinates in each axis) can determine an exact position in the model, which then can be represented by a point (x, y).

Russel's Model:

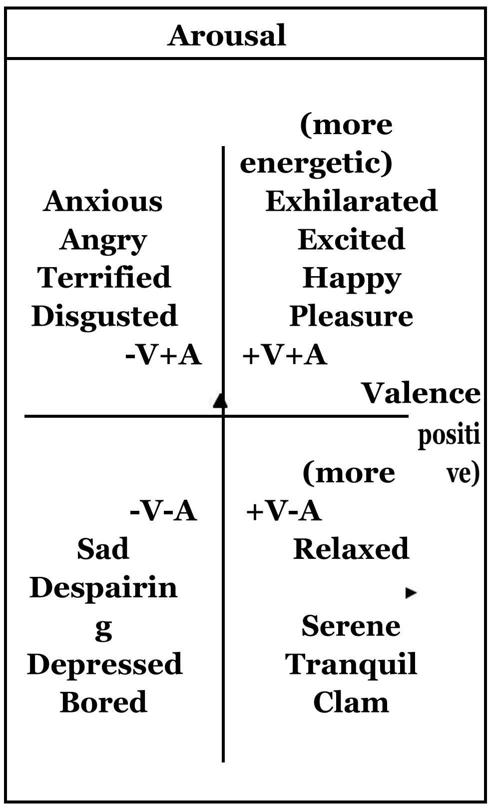


Figure 1. Russell's model of mood

2.2. Mood analysis in MIR research

The present work has as its main goal, as the title suggests, the ability to automatically generate a music playlist according to the mood specified by the user (or, in another perspective, it can be used to generate a playlist to complement the mood of the user – e.g., the user introduces that his mood is anxious, it will be expected that the playlist will include some relaxing music). This selection would be done based on a database consisting of several music files/musical pieces. The pieces that would fit the mood specified by the user (or the pieces that would complement the specified mood) would be added to the final playlist presented to the user.

In particular, this section will analyze the state of art in what concerns to MIR platforms, software tools, feature identification, similarity analysis, playlist creation and algorithm evaluation methods, comparing results and presenting features.

3. The Problem

Although this project operates in a relatively recent field, where the investigation record is still small, there are already some interesting papers and results regarding this subject. The following paragraphs will analyze current investigation in what concerns to mood analysis, similarity and playlist generation, three key concepts in this project.

Mood

As written before, this area, although recent, has already produced new, fresh information. Much is to be found, but also much has already been achieved by now. Some of the papers that inspire this thesis are mentioned in the next paragraphs in what concerns to the mood categories, features and classification used. The results presented in these papers are also discussed.

One of them is "Automatic Mood Detection and Tracking of Music Audio Signals" (Lu, Liu, & Zhang, 2006). It proposes music mood classification with four main clusters: anxious/frantic, depression, contentment and exuberance (based on Thayer's model of mood). In what concerns to features extraction, it proposes the use of three different feature sets: intensity, timbre and rhythm. The paper experiments with two different frameworks for mood detection: hierarchical and non-hierarchical. In both cases, it is used a GMM along with training data. The results presented showed better results for the hierarchical framework.

Another paper (Laar, 2005) discusses and collects information from different papers, regarding different aspects from mood detection in musical pieces. Regarding to mood categories, they range from the four proposed in the paper above to more complex systems (positive/negative affect as one dimension and the pleasantness/unpleasantness versus engagement/disengagement as the other). Another algorithm categorized the songs in two classes (Hostility, Sadness, Guilt and

Love, Excitement, Pride). The paper addresses timbral texture features (centroid, roll off, spectral flux, zero crossings and average silence ratio), tonality coefficient, spectral crest factor, Mel Frequency Cesptral Coefficients (MFCC), Daubechies Wavelet Coefficient Histogram (DWCH), beat and tempo detection, genre information, lyrics, pitch content features, Beats per Minute (BPM) detection and Sum of Absolute Values of the Normalized Fast Fourier Transform (FFT). Classification methods include the use of a neural network with three layers, a GMM (as explained above). The results achieved here showed that there is not “an absolute winner”, ranging from medium to high precision, depending on the relation between granularity (number of categories) and diversity of the music database.

The paper “A Regression Approach to Music Emotion Recognition ”, by Yang et al. (Yang, Lin, Su, & Chen, 2006) addresses mood categories with the innovation of introducing coordinates in the Thayer's model (arousal and valence axis), making it from a continuous perspective. The features extracted were divided into 2 feature sets, one with 114 and another one with a selection of 15 of them. In what concerns to classification, it is addressed in a representative way, and so the system trains 3 different regression algorithms to directly predict arousal and valence values (a, v): multiple linear regression (MLR), support vector regression (SVR) and AdaBoost.RT (BoostR). The results presented in this paper are based in the R2 statistics, where the best combination of data and feature space reaches 58.3% for arousal and 28.1% for valence.

Finally, in the BSc thesis “A Mood-Based Music Classification and Exploration System”, the objective was to analyze not only mood on audio signal of musical pieces but also add mood analysis of the song’s lyrics. The output would be a playlist featuring music with similar mood. The author opted to implement categories based on Russell’s circumplex model of emotion. Features were extracted in conjunction with Hevner’s original mapping of musical features to an emotional space, with slight differences (4 of the 6 features were used – mode, tempo, rhythm and harmony were used, melody and pitch were discarded), plus loudness. Music classification is performed in two steps: a preliminary classification of the song database is made through a decision tree and then it is used the k-NN classifier. The results achieved were reasonable, although the author did not revealed classification statistics: instead, some information about the classification of the music database, lyrics classification, classification vs. music classification experts, classification vs. social tagging services and user evaluation was provided.

Similarity

In what concerns to similarity (basically, the computation of audio and web-based music similarity), two main documents were approached. They are summarized in the next paragraphs.

The tutorial “Music Similarity”, by Elias Pampalk (Pampalk, 2005), refers that music similarity is not only subjective but also context-dependent, with important dimensions such as instrumentation, timbre, melody, harmony, rhythm, tempo, mood, lyrics and social background. The basic schema for this evaluation between two songs consists in the input given – the representation of the song (e. g. Pulse Code Modulation (PCM)), where feature extraction is made. Then the distances between the two songs are computed and compared, using a specified metric (typically Euclidean distance). The tutorial refers experimentation with different features: Zero Crossing Rate (ZCR), MFCC, spectral similarity, fluctuation patterns, chroma complexity and harmony (higher level). The author also points out limitations – or how 100% accuracy is impossible to achieve, because even human experts do not agree always, and some of the aspects pointed out in the article are almost or totally impossible to extract, like sociocultural background, lyrics, mood. It is also pointed out that maybe a perfect similarity measure for applications, at least, is not desirable.

Another paper is "Music Similarity Measures: What's The Use?", by Autocourier et al. (Aucouturier & Pachet, 2002), which proposes a timbral similarity measure (applied to a whole song), based on a Gaussian model of cepstral coefficients (basically, using MFCC modeled with GMM). The application folds in two parts: a timbre extractor (an algorithm that creates a representation of the timbre of a song) and a descriptor, which outputs the proposed timbral similarity measure. The aforementioned timbral measure between two songs can be obtained in two ways: likelihood (matching the probability that the MFCCs of the first song can be generated by the model of the second one, using GMM) or sampling (in the case of not being possible to access the MFCCs of a song while computing the distance, sampling of both GMMs are compared, applying the first method – likelihood – to the mentioned models; Consequently, this method is much more efficient memory-wise). Conducted benchmarking experimentations included the comparison between duplicated songs, songs from the same artist or genre. Then, for each song in the database its timbral distances to all the other songs were computed, comparing these results to textual data (the genre of the titles). These results revealed very poor, and that led to a subjective test, where users were presented with a target song S and 2 songs, A (more closer to S) and B (more distant), and had to order the songs in what concerns to distance to S. The test was performed resorting to ten users and the results showed that about 80% of the songs were well ordered by the application.

4. The Proposed Approach

4.1.AUDIO BASED APPROACH

1. Audio features

In this section all the features addressed in the research conducted throughout three different feature extraction software tools (jAudio, MARSYAS and MIR toolbox) and some papers, which include the ones referenced in the previous part of this section, will be presented and explained with some detail. As mentioned before, this is an extensive list since it is unknown at this point which musical features are going to be used in this project or which ones are dispensable.

Features were divided into four categories: intensity, pitch, rhythm and timbre.

1.1. Intensity

Root Mean Square

Root Mean Square (RMS) returns the power of a signal over a window (McKay, 2005). It can be computed simply by taking the root average of the square of the amplitude, (RMS) (Lartillot)

Root Mean Square derivative
This feature measures the window-to-window change in RMS, serving as an indication of change in signal power (McKay, 2005).

Root Mean Square variability
This feature outputs the standard deviation of the RMS of the last 100 windows (McKay, 2005).

Less-than-average energy
An assessment of the temporal distribution of energy can be obtained through the energy curve, in order to see if it remains constant throughout the signal, or if some frames are more contrastive than others. One way to estimate this consists in computing the low energy rate, *i.e.* the percentage of frames showing less-than-average energy (Lartillot, 2008). Figure 6 shows the visualization of this feature (Lartillot, 2008):

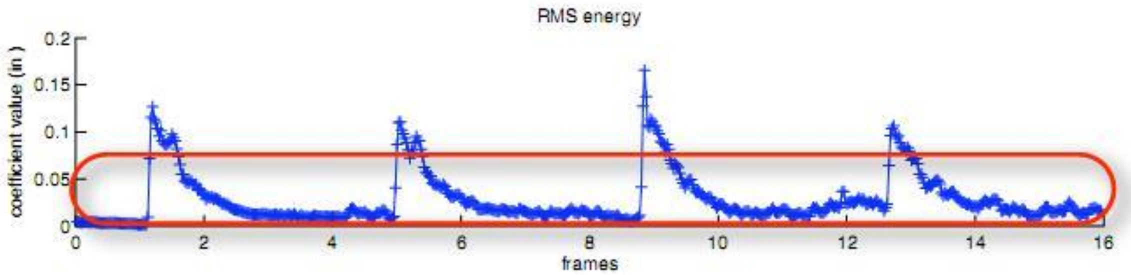


Figure 6: Selected part of the energy curve sows a high value for less-than-average energy feature value (Lartillot, 2008)

Fraction of low energy frames
The fraction of the last 100 windows where the RMS value is less than the mean RMS of the last 100 windows. This can indicate how much of a signal section is quiet relative to the rest of the signal section (McKay, 2005).

1.2. Pitch

Pitch (F0)

Pitch (as an audio feature) typically refers to the fundamental frequency of a monophonic sound signal and can be calculated using various different techniques. It is a subjective property of sound that can be used to order sounds from low to high and is typically related to the fundamental frequency (Tzanetakis, 2002). One of the methods employed in MARSYAS to estimate pitch uses the YIN algorithm, which is based on the autocorrelation method with a number of modifications that combine to prevent errors (de Cheveigné & Kawahara, 2002).

Strongest frequency via FFT Maximum

An estimate of the strongest frequency component of a signal, in Hz, achieved via finding the FFT bin with the highest power (McKay, 2005).

1.3. Rhythm

Beat sum
This feature consists in the sum of all bins in the beat histogram. This is a good measure of the importance of regular beats in a signal (McKay, 2005).

Rhythmic fluctuation
One way to estimate the rhythmic content of a signal is based on spectrogram computation transformed by auditory modeling and then spectrum estimation in each band (Lartillot, 2008). The result of the three phases is illustrated in Figure 7:

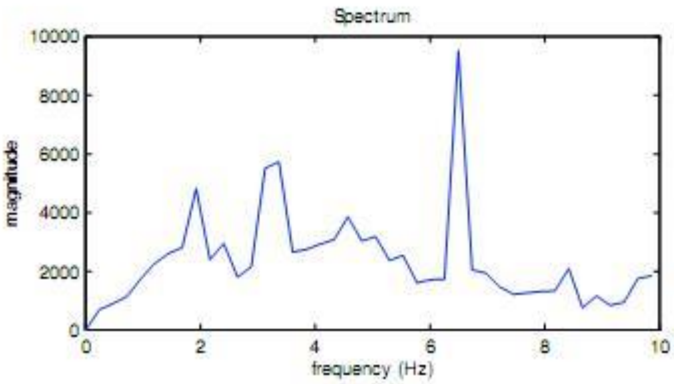


Figure 7: Spectrum summary showing the global repartition of rhythmic periodicities

Strength of strongest beat
This feature measures how strong the strongest beat (the strongest beat in a signal is, in BPM, achieved by finding the highest bin in the beat histogram) is compared to other potential beats (McKay, 2005).

Tempo
This feature consists in the speed (or pace) of a given musical piece - in modern music is indicated in BPM. Its value is estimated by detecting periodicities from the onset detection curve, as exemplified in Figure 8 (Lartillot, 2008):

\

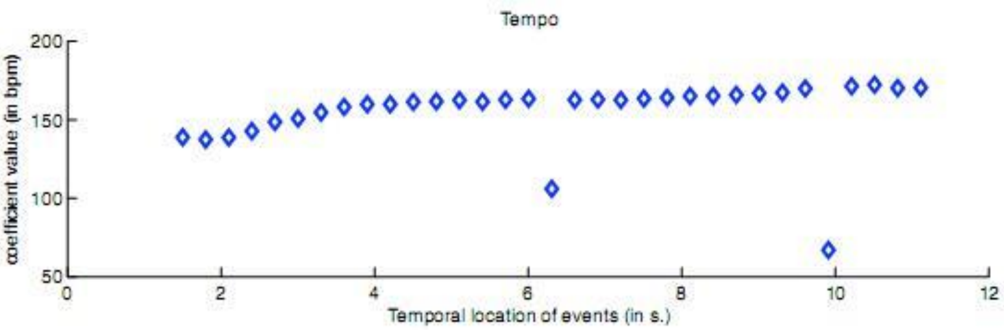


Figure 8: Tempo curve *t* (Lartillot, 2008)

Attack time

Attack time is the estimation of temporal duration for a signal to rise to its peak (e.g., in amplitude). One simple way of describing and compute this feature consists in estimating the attack phase temporal duration

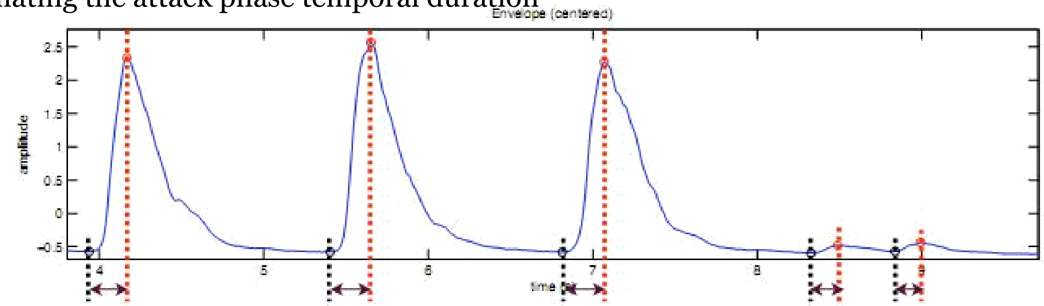


Figure 9: Attack time example through temporal duration (Lartillot, 2008)

Attack slope

This feature can be calculated as a ratio between the magnitude difference at the beginning and the ending of the attack period, and the corresponding time difference (Figure 10) (Lartillot, 2008):

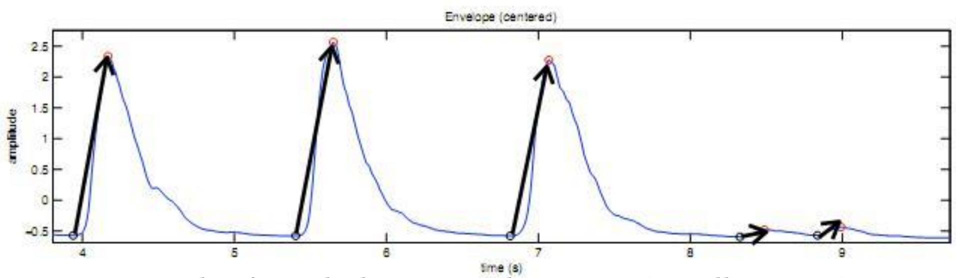


Figure 10: Example of attack slope, given by arrows (Lartillot, 2008)

Spectral roll off

One way to estimate the amount of high frequency in the signal consists in finding the frequency such that a certain fraction of the total energy is contained below that frequency. This ratio is typically fixed by default to 0.85. An example of this is given in Figure 11 (Lartillot, 2008):

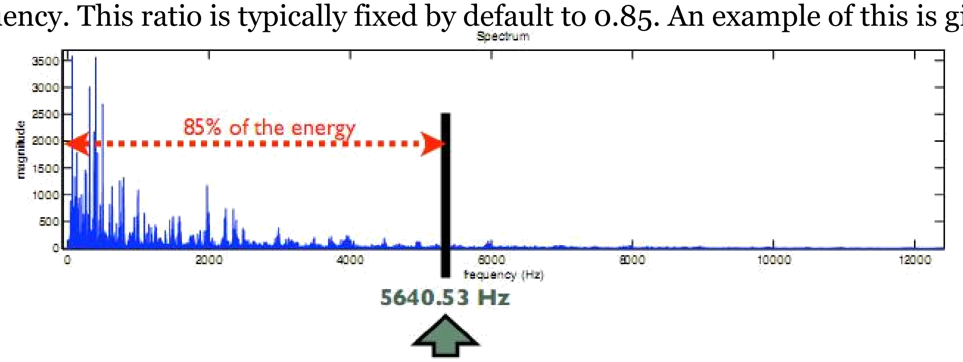


Figure 11: Example of spectral roll off point

High frequency energy (brightness)

This feature is achieved by fixing this time the cut-off frequency, and measuring the amount of energy above that frequency. An example (1500 Hz) is shown in Figure 12:

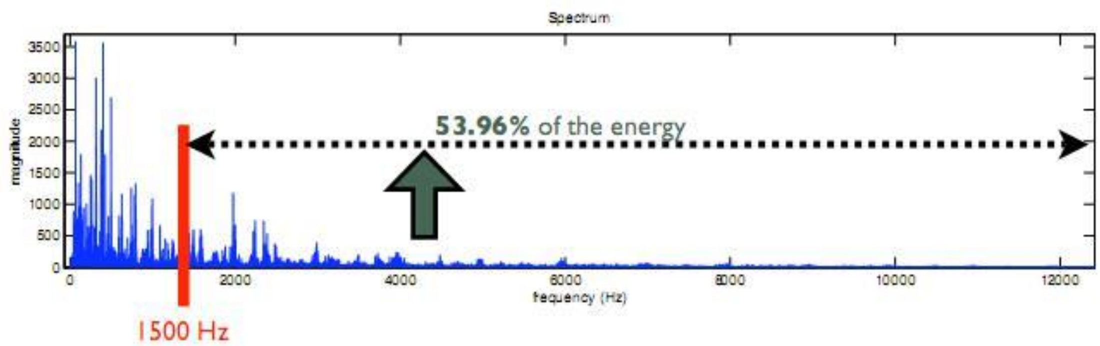
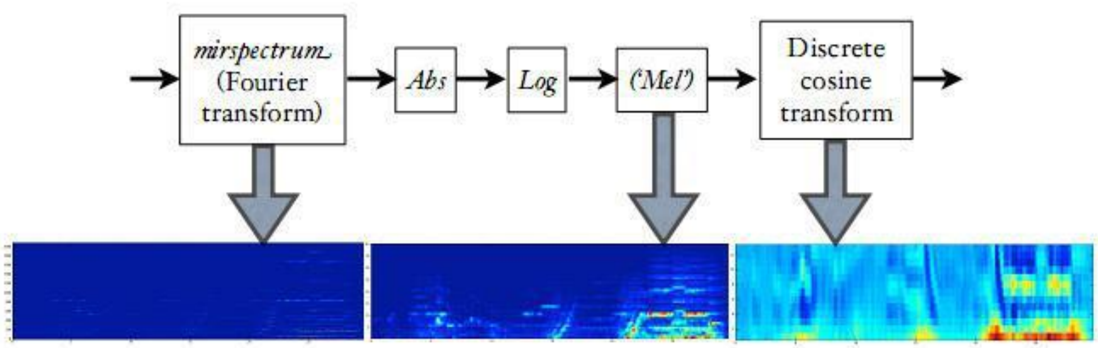


Figure 12: Example of high frequency energy (Lartillot, 2008)

Mel Frequency Cepstral Coefficients

MFCC returns a description of the spectral shape of the sound. The computation of the cepstral follows the scheme present in Figure 13 (Lartillot, 2008)



MFCC are perceptually motivated features that are also based on the Short-time Fourier Transform (STFT). After taking the log-amplitude of the magnitude spectrum, the FFT bins are grouped and smoothed according to the perceptually motivated Mel-frequency scaling. Finally, in order to decorrelate the resulting feature vectors, a Discrete Cosine Transform (DCT) is performed. Although typically 13 coefficients are used for speech representation, it was found that the first five coefficients are adequate for music representation (Tzanetakis, 2002).

Linear Prediction Reflection Coefficients

Linear Prediction Reflection coefficients are used in speech research as an estimate of the speech vocal tract filter (Tzanetakis, 2002). They are also usually used in musical signals.

Sensory dissonance

Also known as roughness, this feature is related to the beating phenomenon whenever a pair of sinusoids is close in frequency. It can be estimated on the frequency ratio of each pair of sinusoids represented in Figure 14 (Lartillot, 2008):

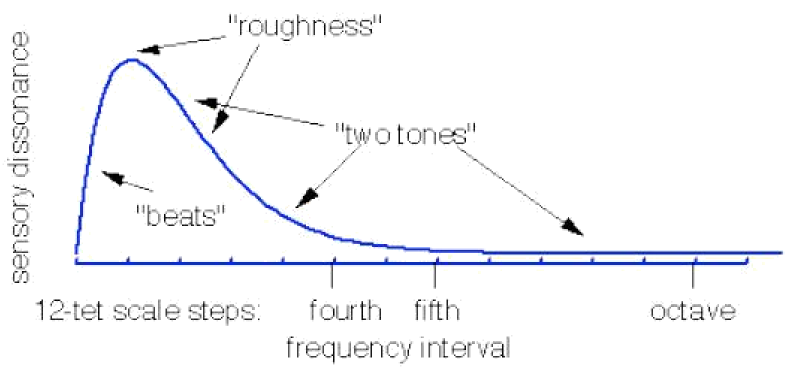


Figure 14: A representation of roughness (Lartillot, 2008)

Spectral centroid

The spectral centroid is defined as the center of gravity of the magnitude spectrum of the STFT.

Mt[n] is the magnitude of the Fourier transform at frame *t* and frequency bin *n*. The centroid is a measure of spectral shape and higher centroid values are related to "brighter" textures with more high frequencies. The spectral centroid has been shown by user experiments to be an important perceptual attribute in the characterization of musical instrument timbre (Tzanetakis, 2002).

Inharmonicity

This feature measures the amount of partials that are not multiples of the fundamental frequency *f0* (Figure 15).

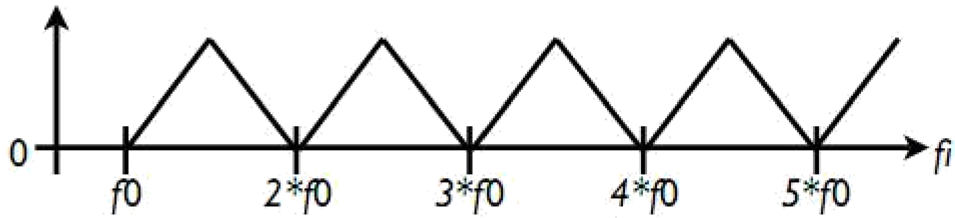


Figure 15: Graphical view for a given fundamental frequency *f0* and its multiples (Lartillot, 2008)

Spectral flux
It is defined as the squared difference between the normalized magnitudes of successive spectral distributions:

Nt[n], *Nt*-1 *[n]* stand for the normalized magnitude of the Fourier transform at the current frame *t*, and the previous frame *t*-1, respectively. This feature is a measure of the amount of local spectral change. It has also been shown by user experiments to be an important perceptual attribute in the characterization of musical instrument timbre (Tzanetakis, 2002).

Strongest frequency via spectral centroid

An estimate of the strongest frequency component of a signal, found via the spectral centroid (McKay, 2005).

Zero-crossing rate
Indicates the number of times the waveform changed sign in a window (the number of times the signal crosses the X-axis), it can be used as an indication of frequency as well as noisiness (McKay, 2005). Figure 16 exemplifies this feature (Lartillot, 2008):

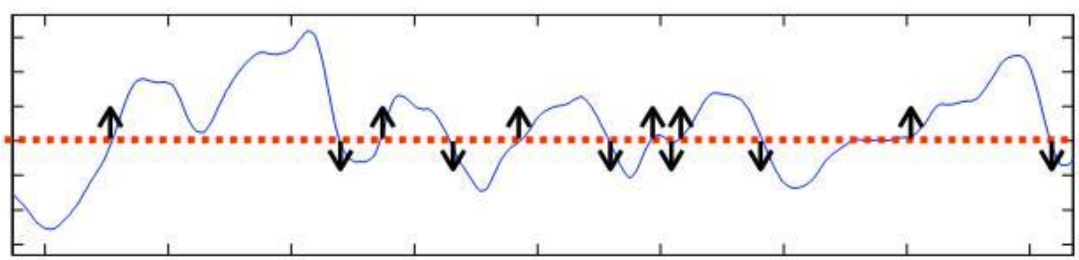


Figure 16: Zero crossing rate example (Lartillot, 2008)

Zero-crossing derivative

This feature can be defined as the absolute value of the window to window change in zero crossings. It can also be considered an indication of change of frequency as well as noisiness (McKay, 2005).

1.5. Tonality

Tonal centroid

The Tonal Centroid is a six-dimensional feature vector based on the Harmonic Network or *Tonnetz*, which is a planar representation of pitch relations where pitch classeshaving close harmonic relations such as fifths, major/minor thirds have smaller Euclidean distances on the plane, represented in Figure 17 (Lee & Slaney, 2007).

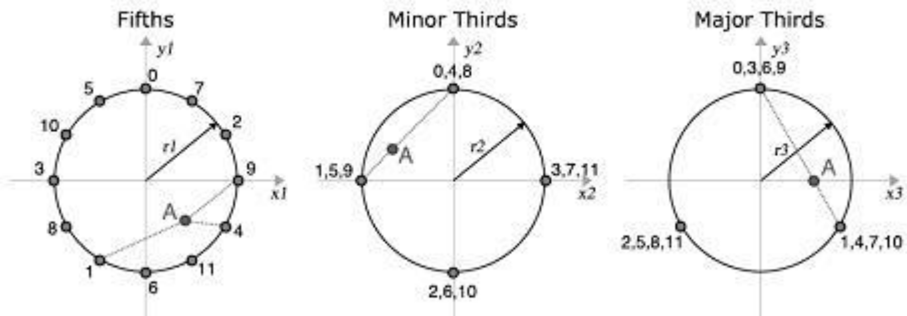


Figure 17: Visualization of the 6-D Tonal Space as three circles: fifths, minor thirds, and major thirds (from left to right) (Lee & Slaney, 2007)

It can be seen in Figure 17 that numbers on the circles correspond to pitch classes and represent nearest neighbors in each circle. Tonal Centroid for A major triad (pitch class 9, 1, and 4) is shown at point A (Lee & Slaney, 2007).

Harmonic change detection function

The Harmonic Change Detection function is the flux of the tonal centroid (Lartillot, 2008).

Key (tonal center positions)

The key feature gives a broad estimation of tonal center positions and their respective clarity, as seen in Figure 18 (Lartillot, 2008)

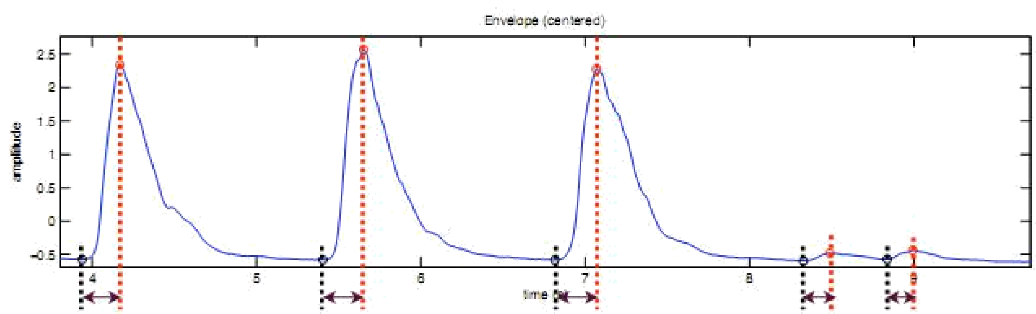


Figure 18: Key graphic example (Lartillot, 2008)

Modality

This feature returns the mode of a key (major or minor, for instance).

2. Musical content features

Musical content features are a set of both rhythmic and pitch content features introduced by George Tzanetakis (Tzanetakis, 2002). This set (and each of the 2 subsets) is based in features extracted previously.

Rhythmic content features
This subset is based on the BH (Beat Histogram) of a song:

- **A0, A1:** relative amplitude (divided by the sum of amplitudes) of the first, and second histogram peak;
- **RA:** ratio of the amplitude of the second peak divided by the amplitude of the first peak;
- **P1, P2:** Period of the first, second peak in BPM;
- **SUM:** overall sum of the histogram (indication of beat strength);

Pitch content features
The following features are computed from the Unfolded Histogram (UPH) and Folded Histogram (FPH) in order to represent pitch content:

- **FA0:** Amplitude of maximum peak of the folded histogram. This corresponds to the most dominant pitch class of the song. For tonal music this peak will typically correspond to the tonic or dominant chord. This peak will be higher for songs that do not have many harmonic changes;
- **UP0:** Period of the maximum peak of the unfolded histogram. This corresponds to the octave range of the dominant musical pitch of the song;

- **FP0:** Period of the maximum peak of the folded histogram. This corresponds to the main pitch class of the song;
- **IPO1:** Pitch interval between the two most prominent peaks of the folded histogram. This corresponds to the main tonal interval relation. For pieces with simple harmonic structure this feature will have value 1 or -1 corresponding to fifth or fourth interval (tonic-dominant);
- **SUM:** The overall sum of the histogram. This is feature is a measure of the strength of the pitch detection;

3. Statistical features

Finally, it is important to underline that is possible to extract statistical information (typically first and second order statistics: e. g. mean and standard deviation) from almost every single feature, as well as higher-order statistics (like skewness, kurtosis, etc.).

This is the main reason for the higher number of features presented by jAudio compared to other frameworks like MARSYAS (as it will be discussed later on this document, sections 2.6.1. jAudio and 2.6.4. Frameworks comparison), since many of them are statistical features obtained from basic features (again, mean and standard deviation, for instance).

4. Support Vector Machines

This classifier uses a support vector algorithm that simply looks for the largest margin (distance) from the separating hyperplane to avoid overfitting as maximum as possible. The support vectors play a key role as the critical elements of the training set. If other training points are changed (or removed) and the training repeated, the same separating hyperplane would be found. All these concepts are visible in Figure 19: SVM example with optimal separating hyperplane Figure 19:

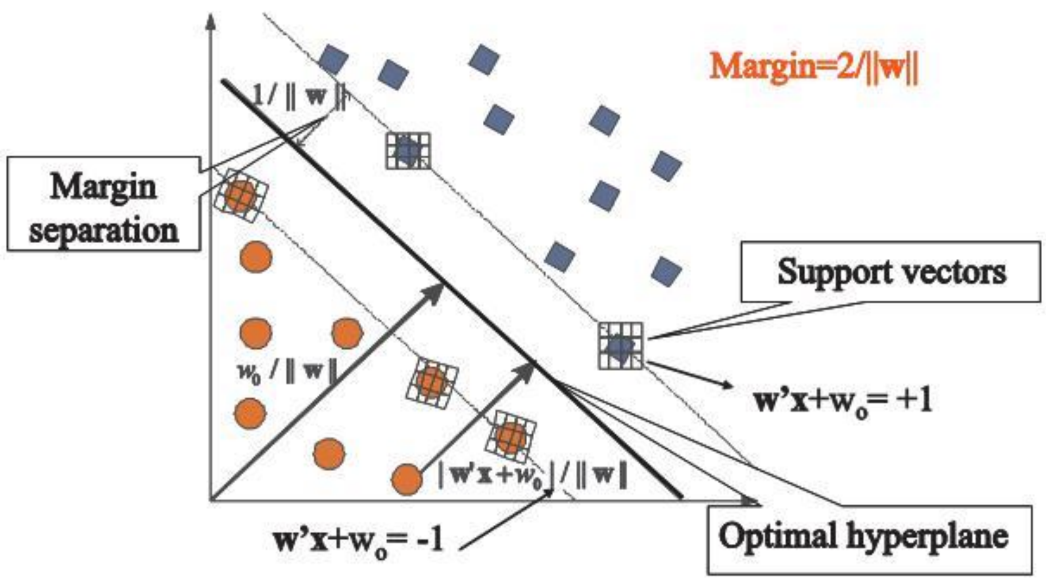


Figure 19: SVM example with optimal separating hyperplane (Ribeiro, 2009)

The separating hyperplane can be linear or non-linear, according to the problem to be solved. The SVM classifier is considered a good solution in what concerns to classification performance achieved / execution time rate, also due to its sophisticated kernel functions and possibility of multiclass classification (Ribeiro, 2009).

5. Euclidean distance in Thayer's mode

This distance metric was proposed in (Yang, Lin, Su, & Chen, 2006) and basically considers that the similarity between two songs can be reached by the Euclidean distance of the songs (points represented as pair of coordinates, x and y) in the Thayer's model, arousal-valence plane.

The Euclidean distance (also known as the Pythagorean metric) is the most common distance metric used, since it is the "ordinary" distance between two points that one would measure with a ruler.

We'll now remember the formula that gives us the Euclidean distance between two points. The Euclidean distance between point *p* and point *q* is the length of the line segment . In Cartesian coordinates, if *p* = (p1, p2 , ..., pn) and *q* = (q1, q2, ..., qn) are two points in Euclidean *n*-space, then the distance from *p* to *q* is given by (Euclidean distance):

6. Existent frameworks and platforms

The most important and known platforms / frameworks available do perform feature extraction are now presented with more detail (jAudio, MARSYAS and MIRtoolbox).

6.1. jAudio

jAudio is a open source feature extraction system that can be used both with a GUI or in command line mode or as a library. jAudio uses a modular plugin interface that avoids core code modification or recompilation when new features are added.

One advantage of jAudio is automated "metafeature" extraction. Metafeatures are template-derived features that can be extracted from one or more other features. Examples of metafeatures implemented in jAudio include Running Mean, Running Standard Deviation and Derivative.

The software includes a wide set of features (28), which can be used with metafeatures and aggregators to expand this number, although a great part of this features are statistical results calculated from other basic features (mean, standard deviation, etc.). However, as mentioned before, this framework (as it is written in Java) is slower and computationally heavy. A screenshot of the application is now presented in Figure 20 (jAudio):

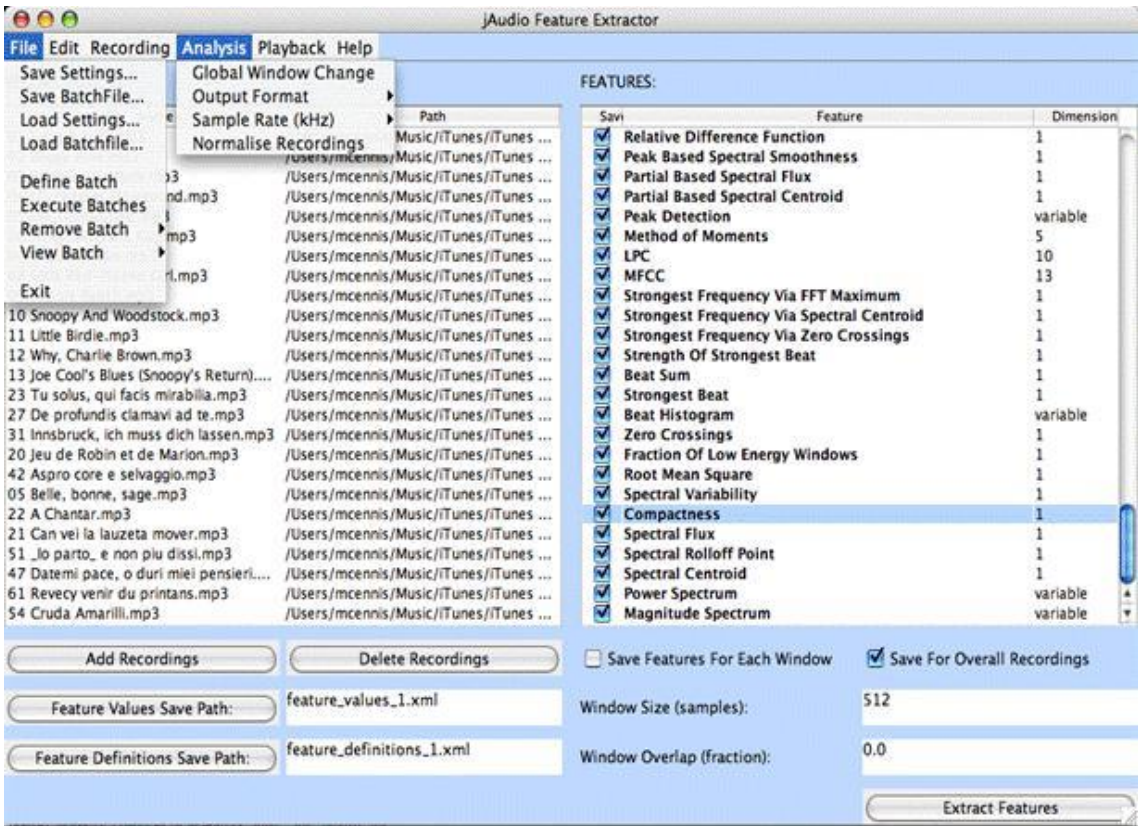


Figure 20: jAudio screenshot, modified artificially to show two menus simultaneously (jAudio)

6.2. MARSYAS

MARSYAS is another open source software framework for audio processing with specific emphasis on MIR applications, being one of the first frameworks built in the area. It has been designed and written by George Tzanetakis, one the most well known and experienced names in the field, with help from students and researchers from around the world. MARSYAS has been used for a variety of projects in both academia and industry (Marsyas).

The basic goal of this framework is to provide a general, extensible and flexible architecture that allows easy experimentation with algorithms and provides fast performance that is useful in developing real time audio analysis and synthesis tools (Overview).

It is written in C++ (with several algorithms) and offers excellent classification times (Audio Music Mood Classification Results), as mentioned earlier. Although it does not come with a GUI it has base support for integration with Qt. These two advantages were crucial for choosing MARSYAS as the framework to be used in this project.

The main disadvantage of this framework is the fact that it apparently has less implemented features than the other two. Although, this flaw can be surpassed with the implementation of the lacking features, if necessary.

4.6.3. MIRtoolbox

MIRtoolbox offers an integrated set of functions written in MATLAB, dedicated to the extraction from audio files of musical features, with the objective of offering an overview of computational approaches in the MIR area. The design is based on a modular framework: the different algorithms are decomposed into stages, formalized using a minimal set of elementary mechanisms. These building blocks form the basic vocabulary of the toolbox, which can then be freely articulated in new original ways. These elementary mechanisms integrate all the different variants proposed by alternative approaches that users can select and parameterize. This synthetic digest of feature extraction tools enables a capitalization of the originality offered by all the alternative strategies. Additionally to the basic computational processes, the toolbox also includes higher-level musical feature extraction tools, whose alternative strategies, and their multiple combinations, can be selected by the user (Lartillot, Toiviainen, & Eerola, Department of Music: MIRtoolbox).

Briefly, it can be said that the main advantages of this toolbox are the integration with MATLAB (which can mean vast documentation, community help and in a sort of way ease of use) and the number of features implemented. However, the MATLAB dependency can also be a disadvantage, mainly because despite the toolbox is open source, MATLAB is not, and so dependency to the application is needed. Also, like jAudio, MATLAB is heavily based and implemented in the Java language, which makes it computationally heavy and slow when it comes to extract features / classify musical pieces.

6.4. Frameworks comparison

The three mentioned frameworks are now compared in what concerns to features, classifiers and distance metrics.

Features

Is now time to make a framework comparison (jAudio, MARSYAS and MIR toolbox and some papers / documentation) in what concerns to implemented features (Table 1).

Feature	Feature class	jAudio	MARSYAS	MIR toolbox	Others
Fraction Of Low Energy Frames	intensity	√			
Less-Than-Average Energy	intensity	√		√	

Root Mean Square Derivative	intensity	√			
Root Mean Square	intensity	√		√	
Root Mean Square Variability	intensity	√			
Pitch (F0)	pitch		√	√	√
Strongest Frequency Via FFT Maximum	pitch	√			
Beat Sum	rhythm	√	√		
Rhythmic Fluctuation	rhythm			√	√
Strength Of Strongest Beat	rhythm	√		√	
Tempo	rhythm	√		√	√
Attack Slope	timbre			√	
Attack Time	timbre			√	
High Frequency Energy (Brightness)	timbre			√	√
Inharmonicity	timbre			√	
Linear Prediction Reflection Coefficients	timbre	√	15		
Mel-Frequency Cepstral Coefficients	timbre	√	√	√	√
Sensory Dissonance	timbre			√	√
Spectral Centroid	timbre	√	√	√	√
Spectral Flux	timbre	√	√	√	√
Spectral Peaks Variability (Irregularity)	timbre	√		√	√
Spectral Roll Off	timbre	√	√	√	√
Strongest Frequency Via Spectral Centroid	timbre	√			
Zero Crossing Rate	timbre	√	√	√	√
Zero Crossing Derivative	timbre	√	√		
Harmonic Change Detection Function	tonality			√	
Key (Tonal Center Positions)	tonality			√	
Modality	tonality			√	
Tonal Centroid	tonality			√	
Musical Content Features	-		√		

Table 1: List of features available in each framework (and papers)

Classification methods

Classfier	jAudio	MARSYAS	MIR toolbox
k-Nearest Neighbor		√	√

Gaussian Mixture Model		√	√
Support Vector Machines		√	

Table 2: Classification methods overview

[15] Linear Prediction Reflection Coefficients are estimated based on Linear Prediction Cepstral Coefficients, which are supported by MARSYAS

Distance metrics

Metric	jAudio	MARSYAS	MIR toolbox	Others
Euclidean distance		√	√	
Manhattan distance			√	
Euclidean distance in Thayer's model16				√
Membership feature vector distance17				√

Table 3: Distance metrics overview

7. Test collection

The musical dataset that is to be used in the application to be developed was kindly provided by Yi-Hsuan Yang, one of the authors of (Yang, Lin, Su, & Chen, 2006), after a personal request. The dataset is the same used in the mentioned paper, and consists of 194 musical pieces from popular japanese, chinese and western albums. Also, two MATLAB .m files (one with the arousal values and the other with valence values for each song in the dataset) were provided.

The musical pieces have common properties: all of the 194 samples have 25 seconds length, and were converted to a uniform format (22,050 Hz, 16 bits, and mono channel PCM WAV) and normalized to the same volume level. The 25 second segment was manually chosen to be representative of the dominant mood in the song (mostly the chorus part) (Yang, Lin, Su, & Chen, 2006).

It is important to notice that this is a provisory dataset, which can be modified and/or expanded in the future if proved necessary.

8. Followed approach

In this project, several approaches were studied and considered, some of them with similar aspects to this application main goal of this application: an automatic playlist generation system, based on mood analysis of a musical database
Options were made in order to achieve this, some by choice, others due to limitations, notably time constraints. As planned, Thayer's mood model, based on a continuous arousal-valence plan, was used.

Also as planned, feature extraction stage was developed using the MARSYAS framework. Although complex, poorly documented and sometimes unstable from version to version, this framework proved to be fast and powerful.

The approach developed in this thesis outputs a "continuous" classification, since it maps each song with a coordinate system (x, y), with values ranging from -1 to 1 in each axis, according to Thayer's model, as pointed out earlier. This implied training a regression model, since it's not a discrete classifier. To achieve this (and since MARSYAS only supports discrete classifiers), an additional classification library was used - libSVM18, in conjunction with MARSYAS.

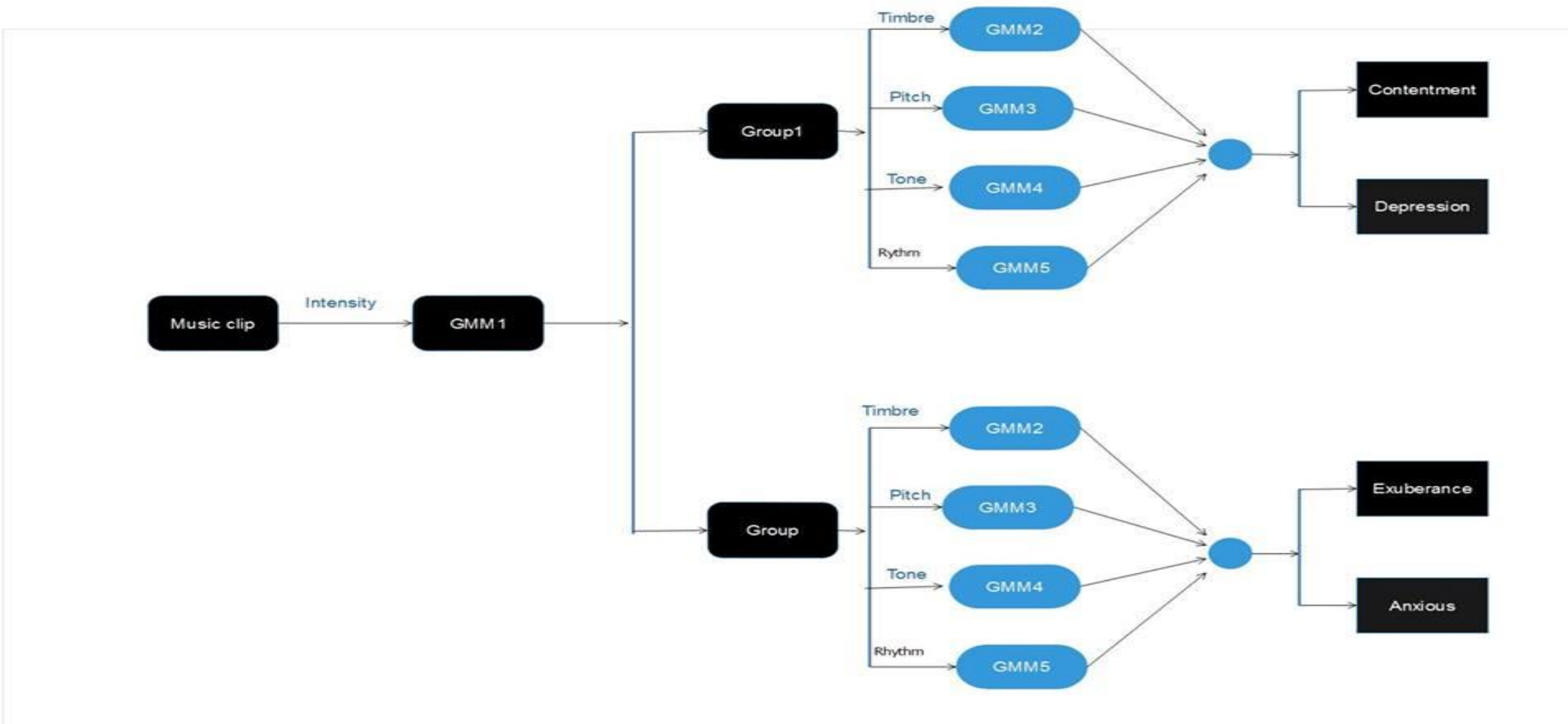
This way, after a train phase, similarity between two songs is calculated based on the Euclidean distance between the songs coordinates (points) in the Thayer's graphic model, as specified earlier in this document. This is done during the test phase and then playlist generation is achieved computing the closer songs to the one used as query. Both annotations and predicted values for arousal and valence coordinates in the test set are compared to evaluate the accuracy of the generated playlists.

- **HEIRARCHICAL MUSIC MOOD DETECTION ALGORITHM**
- Based on Thayer's Mood Model
- Used for classifying a music clip into either of the 4 categories: G1(Exuberance, Anxious),G2(Contentment & depression).
- ***Algorithm:***

1. Start.
2. Convert Music clip into uniform format.
3. Divide Music clip into plurality of frames.
4. Extract Audio features: Spectral features, Beat histogram, Mel-frequency coefficients.
5. Calculate average frame intensities.

6. Classify Music clip into a mood group based on intensity feature.
   a) Determine probabilities of 1st n 2nd group based on intensity.
   b) If P(G1)>P(G2) then select G1.
   Else select G2.

7. Classify Music clip into exact Music mood based on timbral, pitch, tonal & rhythm features.
   a) Determine probabilities of 1st n 2nd group based on intensity.
   b) If P(M1)>P(M2) then select M1
   Else select M2.



4.2.LYRIC BASED APPROACH:

4.2.1 Translating the Words in ANEW

For analyzing the emotion of Chinese song lyrics, an af-fective lexicon called ANCW (Affective Norms for Chi-nese Words) is built from the Bradley's ANEW [4]. The ANEW list was constructed during psycholinguistic exper-iments and contains 1,031 words of all four open classes. As described in it, humans assigned scores to each word according to dimensions such as *pleasure*, *arousal*, and *dominance*. The emotional words in ANEW were trans-lated into Chinese and these constitute the basis of ANCW. 10 people took part in the translation work. Each of them was asked to translate all the words in ANEW into Chi-nese words that he/she thought to be unambiguous and used often in lyrics. The Chinese word that was chosen by the largest number of translators for an ANEW word was picked and added into ANCW. A word may have more than one part of speech(POS), namely performs different functions in different context, and each may have a differ-ent emotion. Therefore, the part of speech of an ANCW word must be indicated. The words, the emotions of which in English culture are different from that in Chinese cul-ture, are simply excluded from ANCW. To see if ANCW is consistent with ANEW, we use Meyers's method [10] to extend ANCW based on a corpus of People's Daily and the extended ANCW includes 18819 words. Meyers extends ANEW to a word list including 73157 words. The distri-butions of the emotion classes of the words in the extended ANCW is illustrated in Figure 3. We find that the emo-tion class distribution of the words in the extended ANCW is similar to the distribution of the words in the extended ANEW. This proves that ANCW is consistent with ANEW and is reasonable.

Table 1. The origins of the words in ANCW

Origin	Translated from ANEW	Synonyms	Added by lyrics corpus
# of words	985	2995	71

4.2.2. Extending ANEW

However, the words translated from ANEW are not suffi-cient for the purpose of detecting emotions of lyrics so it is necessary to extend ANCW. We extend ANCW in two ways. In one way, with each word in ANCW as a seed, we find out all of its synonyms in TONG YI CI CI LIN [2] . Then, only synonyms with the same part of speech as that of their seed are added to ANCW. In the other way, we extract all constructions of apposition and coordination in a corpus of lyrics(containing 1800 lyrics) by an off-the-shelf natural language processing tool [8]. If either word in such a construction is in ANCW, its counterpart is added to ANCW. The origins of the words in ANCW is shown in Table 1 and valence-arousal distribution of the words in ANCW is illustrated in Figure 4. To indi-cate whether a word in ANCW is a translated word from ANEW or a later added word, we attach an *origin* property to each word. Therefore, terms in the affect lexicon have the following form: *< word, origin, POS, valence, arousal >*

Affective Norms for English Words. **All Subjects** *Table 1*

Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
abduction	621	2.76	(2.06)	5.53	(2.43)	3.49	(2.38)	1
abortion	622	3.50	(2.30)	5.39	(2.80)	4.59	(2.54)	6
absurd	623	4.26	(1.82)	4.36	(2.20)	4.73	(1.72)	17
abundance	624	6.59	(2.01)	5.51	(2.63)	5.80	(2.16)	13
abuse	1	1.80	(1.23)	6.83	(2.70)	3.69	(2.94)	18
acceptance	625	7.98	(1.42)	5.40	(2.70)	6.64	(1.91)	49
accident	2	2.05	(1.19)	6.26	(2.87)	3.76	(2.22)	33
ace	626	6.88	(1.93)	5.50	(2.66)	6.39	(2.31)	15
ache	627	2.46	(1.52)	5.00	(2.45)	3.54	(1.73)	4
achievement	3	7.89	(1.38)	5.53	(2.81)	6.56	(2.35)	65
activate	4	5.46	(0.98)	4.86	(2.56)	5.43	(1.84)	2
addict	581	2.48	(2.08)	5.66	(2.26)	3.72	(2.54)	1
addicted	628	2.51	(1.42)	4.81	(2.46)	3.46	(2.23)	3
admired	5	7.74	(1.84)	6.11	(2.36)	7.53	(1.94)	17
adorable	6	7.81	(1.24)	5.12	(2.71)	5.74	(2.48)	3
adult	546	6.49	(1.50)	4.76	(1.95)	5.75	(2.21)	25
advantage	629	6.95	(1.85)	4.76	(2.18)	6.36	(2.23)	73
adventure	630	7.60	(1.50)	6.98	(2.15)	6.46	(1.67)	14
affection	7	8.39	(0.86)	6.21	(2.75)	6.08	(2.22)	18
afraid	8	2.00	(1.28)	6.67	(2.54)	3.98	(2.63)	57
aggressive	9	5.10	(1.68)	5.83	(2.33)	5.59	(2.40)	17
agility	22	6.46	(1.57)	4.85	(1.80)	5.87	(1.52)	3
agony	10	2.43	(2.17)	6.06	(2.67)	4.02	(2.49)	9
agreement	631	7.08	(1.59)	5.02	(2.24)	6.22	(1.85)	106
air	632	6.34	(1.56)	4.12	(2.30)	5.10	(1.56)	257
alcoholic	582	2.84	(2.34)	5.69	(2.36)	4.45	(2.56)	3
alert	11	6.20	(1.76)	6.85	(2.53)	5.96	(2.24)	33
alien	633	5.60	(1.82)	5.45	(2.15)	4.64	(2.07)	16
alimony	634	3.95	(2.00)	4.30	(2.29)	4.63	(2.30)	2
alive	635	7.25	(2.22)	5.50	(2.74)	6.39	(2.15)	57
allergy	636	3.07	(1.64)	4.64	(2.34)	3.21	(1.77)	1
alley	637	4.48	(1.97)	4.91	(2.42)	4.00	(1.70)	8
alone	12	2.41	(1.77)	4.83	(2.66)	3.70	(2.42)	195
aloof	13	4.90	(1.92)	4.28	(2.10)	4.69	(1.92)	5
ambition	14	7.04	(1.98)	5.61	(2.92)	6.93	(2.07)	19
ambulance	15	2.47	(1.50)	7.33	(1.96)	3.22	(2.29)	6
angel	16	7.53	(1.58)	4.83	(2.63)	4.97	(2.34)	18
anger	17	2.34	(1.32)	7.63	(1.91)	5.50	(2.82)	48
angry	18	2.85	(1.70)	7.17	(2.07)	5.55	(2.74)	45
anguished	19	2.12	(1.56)	5.33	(2.69)	3.45	(2.37)	2
ankle	638	5.27	(1.54)	4.16	(2.03)	4.77	(1.74)	8
annoy	20	2.74	(1.81)	6.49	(2.17)	5.09	(2.04)	2
answer	639	6.63	(1.68)	5.41	(2.43)	5.85	(1.88)	152
anxious	21	4.81	(1.98)	6.92	(1.81)	5.33	(1.82)	29
applause	640	7.50	(1.50)	5.80	(2.79)	6.48	(2.11)	14
appliance	641	5.10	(1.21)	4.05	(2.06)	5.05	(1.34)	5
arm	642	5.34	(1.82)	3.59	(2.40)	5.07	(1.50)	94
army	23	4.72	(1.75)	5.03	(2.03)	5.03	(2.45)	132
aroused	24	7.97	(1.00)	6.63	(2.70)	6.14	(1.97)	20
arrogant	25	3.69	(2.40)	5.65	(2.23)	5.14	(2.71)	2
art	643	6.68	(2.10)	4.86	(2.88)	5.30	(2.33)	208
assassin	26	3.09	(2.09)	6.28	(2.53)	4.33	(2.68)	6
assault	27	2.03	(1.55)	7.51	(2.28)	3.94	(3.10)	15
astonished	28	6.56	(1.61)	6.58	(2.22)	5.16	(1.79)	6
astronaut	501	6.66	(1.60)	5.28	(2.11)	5.20	(1.95)	2
athletics	644	6.61	(2.08)	6.10	(2.29)	6.12	(2.12)	9
autumn	29	6.30	(2.14)	4.51	(2.50)	5.15	(1.85)	22
avalanche	645	3.29	(1.95)	5.54	(2.37)	3.61	(2.00)	1
avenue	646	5.50	(1.37)	4.12	(2.01)	5.40	(1.53)	46
awed	30	6.70	(1.38)	5.74	(2.31)	5.30	(2.03)	5
baby	31	8.22	(1.20)	5.53	(2.80)	5.00	(2.80)	62
bake	647	6.17	(1.71)	5.10	(2.30)	5.49	(1.88)	12
bandage	648	4.54	(1.75)	3.90	(2.07)	4.52	(1.89)	4
bankrupt	32	2.00	(1.31)	6.21	(2.79)	3.27	(2.39)	5
banner	649	5.40	(0.83)	3.83	(1.95)	4.80	(1.57)	8
bar	650	6.42	(2.05)	5.00	(2.83)	5.47	(1.94)	82
barrel	651	5.05	(1.46)	3.36	(2.28)	4.89	(1.57)	24
basket	547	5.45	(1.15)	3.63	(2.02)	5.76	(1.45)	17
bastard	33	3.36	(2.16)	6.07	(2.15)	4.17	(2.40)	12
bath	502	7.33	(1.45)	4.16	(2.31)	6.41	(1.87)	26
bathroom	548	5.55	(1.36)	3.88	(1.72)	5.65	(1.59)	18
bathtub	652	6.69	(1.57)	4.36	(2.59)	5.76	(1.76)	4
beach	34	8.03	(1.59)	5.53	(3.07)	5.44	(2.52)	61
beast	653	4.23	(2.41)	5.57	(2.61)	4.89	(2.29)	7
beautiful	654	7.60	(1.64)	6.17	(2.34)	6.29	(1.81)	127
beauty	35	7.82	(1.16)	4.95	(2.57)	5.53	(2.10)	71
bed	549	7.51	(1.38)	3.61	(2.56)	6.88	(1.78)	127
bees	583	3.20	(2.07)	6.51	(2.14)	4.16	(2.11)	15

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)	Arousal Mean(SD)	Dominance Mean (SD)	Word Frequency
beggar	36	3.22 (2.02)	4.91 (2.45)	4.09 (2.38)	2
bench	655	4.61 (1.40)	3.59 (2.07)	4.68 (1.38)	35
bereavement	656	4.57 (1.70)	4.20 (2.15)	4.33 (1.73)	4
betray	37	1.68 (1.02)	7.24 (2.06)	4.92 (2.97)	4
beverage	657	6.83 (1.48)	5.21 (2.46)	5.63 (2.17)	5
bird	38	7.27 (1.36)	3.17 (2.23)	4.42 (2.26)	31
birthday	39	7.84 (1.92)	6.68 (2.11)	5.89 (2.61)	18
black	543	5.39 (1.80)	4.61 (2.24)	5.14 (1.79)	203
blackmail	40	2.95 (1.95)	6.03 (2.70)	3.54 (2.67)	2
bland	658	4.10 (1.08)	3.29 (1.89)	4.88 (1.27)	3
blase	41	4.89 (1.16)	3.94 (1.76)	4.57 (1.44)	7
blasphemy	659	3.75 (2.26)	4.93 (2.34)	4.75 (1.59)	4
bless	42	7.19 (1.69)	4.05 (2.59)	5.52 (2.22)	9
blind	43	3.05 (1.99)	4.39 (2.36)	3.28 (1.91)	47
bliss	660	6.95 (2.24)	4.41 (2.95)	6.12 (2.15)	4
blister	661	2.88 (1.75)	4.10 (2.34)	3.98 (1.90)	3
blond	662	6.43 (2.04)	5.07 (2.70)	5.74 (1.67)	11
bloody	584	2.90 (1.98)	6.41 (2.00)	3.96 (1.89)	8
blossom	44	7.26 (1.18)	5.03 (2.65)	5.53 (2.21)	7
blubber	663	3.52 (1.99)	4.57 (2.38)	3.86 (1.97)	1
blue	544	6.76 (1.78)	4.31 (2.20)	5.63 (1.64)	143
board	664	4.82 (1.23)	3.36 (2.12)	4.98 (1.77)	239
body	665	5.55 (2.37)	5.52 (2.63)	5.34 (2.12)	276
bold	45	6.80 (1.61)	5.60 (2.21)	6.67 (1.81)	21
bomb	46	2.10 (1.19)	7.15 (2.40)	4.54 (2.88)	36
book	47	5.72 (1.54)	4.17 (2.49)	5.30 (2.05)	193
bored	48	2.95 (1.35)	2.83 (2.31)	4.11 (1.70)	14
bottle	666	6.15 (1.49)	4.79 (2.44)	4.78 (1.65)	76
bouquet	667	7.02 (1.84)	5.46 (2.47)	6.15 (1.80)	4
bowl	49	5.33 (1.33)	3.47 (2.12)	4.69 (1.67)	23
boxer	585	5.51 (1.80)	5.12 (2.26)	5.10 (1.64)	1
boy	50	6.32 (1.60)	4.58 (2.37)	5.34 (2.20)	242
brave	668	7.15 (1.64)	6.15 (2.45)	7.22 (1.86)	24
breast	51	6.50 (1.78)	5.37 (2.39)	5.39 (2.27)	11
breeze	669	6.85 (1.71)	4.37 (2.32)	5.54 (1.67)	14
bride	670	7.34 (1.71)	5.55 (2.74)	5.74 (2.36)	33
bright	671	7.50 (1.55)	5.40 (2.33)	6.34 (1.82)	87
broken	672	3.05 (1.92)	5.43 (2.42)	4.14 (1.62)	63
brother	52	7.11 (2.17)	4.71 (2.68)	5.12 (2.31)	73
brutal	53	2.80 (1.90)	6.60 (2.36)	4.59 (2.70)	7
building	550	5.29 (1.15)	3.92 (1.94)	5.25 (1.57)	160
bullet	673	3.29 (2.06)	5.33 (2.48)	3.90 (2.61)	28
bunny	54	7.24 (1.32)	4.06 (2.61)	4.97 (2.18)	1
burdened	55	2.50 (1.32)	5.63 (2.07)	5.03 (2.35)	4
burial	56	2.05 (1.41)	5.08 (2.40)	3.55 (1.95)	11
burn	586	2.73 (1.72)	6.22 (1.91)	4.22 (1.83)	15
bus	541	4.51 (1.57)	3.55 (1.80)	4.84 (1.75)	34
busybody	674	5.17 (2.02)	4.84 (2.41)	5.45 (1.97)	.
butter	57	5.33 (1.20)	3.17 (1.84)	4.67 (1.69)	27
butterfly	58	7.17 (1.20)	3.47 (2.39)	4.65 (2.27)	2
cabinet	675	5.05 (0.31)	3.43 (1.85)	4.73 (1.66)	17
cake	59	7.26 (1.27)	5.00 (2.37)	5.16 (2.05)	9
cancer	60	1.50 (0.85)	6.42 (2.83)	3.42 (2.99)	25
candy	61	6.54 (2.09)	4.58 (2.40)	5.33 (1.91)	16
cane	677	4.00 (1.80)	4.20 (1.93)	4.27 (1.95)	12
cannon	678	4.90 (2.20)	4.71 (2.84)	5.17 (2.29)	7
capable	62	7.16 (1.39)	5.08 (2.07)	6.47 (1.94)	66
car	551	7.73 (1.63)	6.24 (2.04)	6.98 (2.06)	274
carcass	679	3.34 (1.92)	4.83 (2.07)	4.90 (1.79)	7
carefree	63	7.54 (1.38)	4.17 (2.84)	5.78 (2.50)	9
caress	64	7.84 (1.16)	5.14 (3.00)	5.83 (2.13)	1
cash	503	8.37 (1.00)	7.37 (2.21)	6.96 (2.39)	36
casino	680	6.81 (1.66)	6.51 (2.12)	5.12 (2.15)	2
cat	504	5.72 (2.43)	4.38 (2.24)	6.16 (2.05)	.
cell	587	3.82 (1.70)	4.08 (2.19)	4.12 (2.13)	65
cellar	681	4.32 (1.68)	4.39 (2.33)	4.66 (1.61)	26
cemetery	65	2.63 (1.40)	4.82 (2.66)	4.27 (2.14)	15
chair	66	5.08 (0.98)	3.15 (1.77)	4.56 (1.60)	66
champ	682	7.18 (1.97)	6.00 (2.43)	6.77 (2.00)	1
champion	67	8.44 (0.90)	5.85 (3.15)	6.50 (2.85)	23
chance	683	6.02 (1.77)	5.38 (2.58)	4.64 (1.93)	131
chaos	684	4.17 (2.36)	6.67 (2.06)	3.86 (1.95)	17
charm	68	6.77 (1.58)	5.16 (2.25)	5.57 (2.25)	26
cheer	69	8.10 (1.17)	6.12 (2.45)	6.00 (2.06)	8
child	70	7.08 (1.98)	5.55 (2.29)	5.10 (2.30)	213
chin	685	5.29 (1.27)	3.31 (1.98)	5.26 (1.48)	27
chocolate	505	6.88 (1.89)	5.29 (2.55)	5.18 (1.97)	9
christmas	686	7.80 (1.55)	6.27 (2.56)	5.37 (2.09)	27

Affective Norms for English Words. **All Subjects** *Table 1*

Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)	Arousal Mean(SD)	Dominance Mean (SD)	Word Frequency
church	71	6.28 (2.31)	4.34 (2.45)	5.00 (2.42)	348
circle	687	5.67 (1.26)	3.86 (2.13)	5.03 (1.46)	60
circus	72	7.30 (1.84)	5.97 (2.59)	5.39 (2.25)	7
city	73	6.03 (1.37)	5.24 (2.53)	5.74 (2.08)	393
cliff	553	4.67 (2.08)	6.25 (2.15)	4.35 (2.11)	11
clock	688	5.14 (1.54)	4.02 (2.54)	4.67 (1.97)	20
clothing	74	6.54 (1.85)	4.78 (2.88)	5.33 (2.14)	20
clouds	533	6.18 (2.18)	3.30 (2.08)	5.22 (1.66)	38
clumsy	689	4.00 (2.22)	5.18 (2.40)	3.86 (1.79)	6
coarse	690	4.55 (1.42)	4.21 (1.84)	5.00 (1.43)	10
coast	691	5.98 (1.86)	4.59 (2.31)	5.67 (1.71)	61
cockroach	75	2.81 (2.11)	6.11 (2.78)	4.74 (2.58)	2
coffin	76	2.56 (1.96)	5.03 (2.79)	4.08 (2.54)	7
coin	692	6.02 (1.96)	4.29 (2.48)	5.66 (1.68)	10
cold	693	4.02 (1.99)	5.19 (2.23)	4.69 (1.73)	171
color	694	7.02 (1.57)	4.73 (2.64)	6.17 (1.82)	141
column	695	5.17 (0.85)	3.62 (1.91)	4.81 (1.58)	71
comedy	77	8.37 (0.94)	5.85 (2.81)	5.44 (2.08)	39
comfort	696	7.07 (2.14)	3.93 (2.85)	5.70 (2.05)	43
computer	552	6.24 (1.61)	4.75 (1.93)	5.29 (1.99)	13
concentrate	78	5.20 (1.28)	4.65 (2.13)	4.97 (1.75)	11
confident	79	7.98 (1.29)	6.22 (2.41)	7.68 (1.94)	16
confused	80	3.21 (1.51)	6.03 (1.88)	4.24 (1.91)	44
consoled	81	5.78 (1.64)	4.53 (2.22)	4.44 (1.84)	2
contempt	82	3.85 (2.13)	5.28 (2.04)	5.13 (1.73)	15
contents	83	4.89 (0.89)	4.32 (2.14)	4.85 (1.49)	16
context	84	5.20 (1.38)	4.22 (2.24)	5.17 (1.39)	2
controlling	85	3.80 (2.25)	6.10 (2.19)	5.17 (3.15)	23
cook	697	6.16 (1.89)	4.44 (1.96)	5.14 (1.49)	47
cord	698	5.10 (1.09)	3.54 (2.09)	5.00 (1.22)	6
cork	699	5.22 (1.13)	3.80 (2.18)	4.98 (1.04)	9
corner	700	4.36 (1.21)	3.91 (1.92)	4.12 (1.66)	115
corpse	86	2.18 (1.48)	4.74 (2.94)	3.59 (2.44)	7
corridor	701	4.88 (1.14)	3.63 (2.41)	5.00 (1.48)	17
corrupt	702	3.32 (2.32)	4.67 (2.35)	4.64 (2.30)	8
cottage	87	6.45 (1.52)	3.39 (2.54)	5.39 (1.78)	19
couple	506	7.41 (1.97)	6.39 (2.31)	6.02 (2.28)	122
cow	554	5.57 (1.53)	3.49 (2.13)	5.32 (1.61)	29
coward	703	2.74 (1.64)	4.07 (2.19)	2.83 (1.61)	8
cozy	88	7.39 (1.53)	3.32 (2.28)	4.89 (2.28)	1
crash	89	2.31 (1.44)	6.95 (2.44)	3.44 (2.21)	20
crime	704	2.89 (2.06)	5.41 (2.69)	4.12 (2.24)	34
criminal	705	2.93 (1.66)	4.79 (2.51)	3.34 (1.73)	24
crisis	706	2.74 (2.23)	5.44 (3.07)	3.60 (2.47)	82
crown	90	6.58 (1.42)	4.28 (2.53)	6.06 (2.15)	19
crucify	91	2.23 (1.72)	6.47 (2.47)	3.74 (2.48)	2
crude	707	3.12 (1.65)	5.07 (2.37)	4.27 (1.94)	15
cruel	92	1.97 (1.67)	5.68 (2.65)	4.24 (2.84)	15
crushed	93	2.21 (1.74)	5.52 (2.87)	3.36 (2.69)	10
crutch	708	3.43 (1.62)	4.14 (2.05)	3.91 (1.79)	1
cuddle	94	7.72 (1.92)	4.40 (2.67)	5.85 (2.42)	.
cuisine	709	6.64 (1.48)	4.39 (1.99)	5.41 (1.19)	1
curious	95	6.08 (1.63)	5.82 (1.64)	5.42 (1.60)	46
curtains	710	4.83 (0.83)	3.67 (1.83)	5.05 (1.56)	8
custom	96	5.85 (1.53)	4.66 (2.12)	5.00 (1.87)	14
cut	711	3.64 (2.08)	5.00 (2.32)	4.70 (1.98)	192
cute	97	7.62 (1.01)	5.53 (2.71)	4.86 (2.32)	5
cyclone	98	3.60 (2.38)	6.36 (2.89)	4.89 (2.56)	.
dagger	99	3.38 (1.77)	6.14 (2.64)	4.52 (2.27)	1
damage	712	3.05 (1.65)	5.57 (2.26)	3.88 (1.86)	33
dancer	507	7.14 (1.56)	6.00 (2.20)	6.02 (1.93)	31
danger	713	2.95 (2.22)	7.32 (2.07)	3.59 (2.31)	70
dark	714	4.71 (2.36)	4.28 (2.21)	4.84 (2.15)	185
dawn	715	6.16 (2.33)	4.39 (2.81)	5.16 (2.23)	28
daylight	716	6.80 (2.17)	4.77 (2.50)	5.48 (2.14)	15
dazzle	717	7.29 (1.09)	6.33 (2.02)	5.62 (1.81)	1
dead	588	1.94 (1.76)	5.73 (2.73)	2.84 (2.32)	174
death	100	1.61 (1.40)	4.59 (3.07)	3.47 (2.50)	277
debt	101	2.22 (1.17)	5.68 (2.74)	3.02 (2.16)	13
deceit	718	2.90 (1.63)	5.68 (2.46)	3.95 (2.12)	2
decompose	102	3.20 (1.81)	4.65 (2.39)	4.02 (1.91)	1
decorate	719	6.93 (1.30)	5.14 (2.39)	6.05 (1.86)	2
defeated	103	2.34 (1.66)	5.09 (3.00)	3.11 (2.34)	15
defiant	104	4.26 (2.12)	6.10 (2.51)	5.77 (2.40)	3
deformed	720	2.41 (1.66)	4.07 (2.34)	3.95 (2.18)	.
delayed	721	3.07 (1.74)	5.62 (2.39)	3.64 (1.94)	25
delight	105	8.26 (1.04)	5.44 (2.88)	5.79 (2.24)	29
demon	106	2.11 (1.56)	6.76 (2.68)	4.89 (2.89)	9

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
dentist	589	4.02	(2.23)	5.73	(2.13)	3.80	(2.16)	12
depressed	107	1.83	(1.42)	4.72	(2.95)	2.74	(2.13)	11
depression	108	1.85	(1.67)	4.54	(3.19)	2.91	(2.27)	24
derelict	722	4.28	(1.84)	4.10	(1.94)	4.78	(1.56)	1
deserter	109	2.45	(1.80)	5.50	(2.55)	3.77	(2.29)	.
desire	508	7.69	(1.39)	7.35	(1.76)	6.49	(1.83)	79
despairing	110	2.43	(1.47)	5.68	(2.37)	3.43	(2.11)	4
despise	111	2.03	(1.38)	6.28	(2.43)	4.72	(2.80)	7
destroy	112	2.64	(2.03)	6.83	(2.38)	4.94	(2.86)	48
destruction	723	3.16	(2.44)	5.82	(2.71)	3.93	(2.29)	38
detached	113	3.86	(1.88)	4.26	(2.57)	3.63	(2.15)	12
detail	724	5.55	(1.58)	4.10	(2.24)	5.21	(1.60)	72
detest	114	2.17	(1.30)	6.06	(2.39)	5.83	(2.60)	1
devil	115	2.21	(1.99)	6.07	(2.61)	5.35	(2.75)	25
devoted	116	7.41	(1.37)	5.23	(2.21)	6.18	(2.36)	51
diamond	117	7.92	(1.20)	5.53	(2.96)	5.54	(2.28)	8
dignified	118	7.10	(1.26)	4.12	(2.29)	6.12	(2.40)	7
dinner	509	7.16	(1.50)	5.43	(2.14)	6.10	(1.87)	91
diploma	119	8.00	(1.39)	5.67	(2.80)	6.76	(2.50)	.
dirt	725	4.17	(1.77)	3.76	(2.26)	4.83	(1.82)	43
dirty	590	3.08	(2.05)	4.88	(2.29)	4.70	(2.12)	36
disappoint	120	2.39	(1.44)	4.92	(2.64)	3.29	(2.32)	.
disaster	121	1.73	(1.13)	6.33	(2.70)	3.52	(2.42)	26
discomfort	726	2.19	(1.23)	4.17	(2.44)	3.86	(2.26)	7
discouraged	122	3.00	(2.16)	4.53	(2.11)	3.61	(2.01)	15
disdainful	123	3.68	(1.90)	5.04	(2.14)	4.55	(1.92)	2
disgusted	124	2.45	(1.41)	5.42	(2.59)	4.34	(1.94)	6
disloyal	125	1.93	(1.61)	6.56	(2.21)	3.79	(2.75)	2
displeased	126	2.79	(2.23)	5.64	(2.48)	4.19	(2.19)	7
distressed	127	1.94	(1.10)	6.40	(2.38)	3.76	(2.41)	4
disturb	727	3.66	(2.00)	5.80	(2.39)	4.55	(1.90)	10
diver	510	6.45	(1.55)	5.04	(2.10)	5.04	(1.91)	1
divorce	128	2.22	(1.88)	6.33	(2.71)	3.26	(2.24)	29
doctor	129	5.20	(2.54)	5.86	(2.70)	4.89	(2.75)	100
dog	511	7.57	(1.66)	5.76	(2.50)	6.25	(2.10)	75
doll	728	6.09	(1.96)	4.24	(2.43)	4.61	(2.07)	10
dollar	729	7.47	(1.72)	6.07	(2.67)	6.33	(2.42)	46
door	130	5.13	(1.44)	3.80	(2.29)	4.69	(1.72)	312
dove	730	6.90	(1.54)	3.79	(2.28)	5.48	(1.70)	4

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
dreadful	131	2.26	(1.91)	5.84	(2.62)	4.10	(2.36)	10
dream	132	6.73	(1.75)	4.53	(2.72)	5.53	(1.98)	64
dreary	731	3.05	(1.58)	2.98	(2.18)	3.81	(1.64)	6
dress	133	6.41	(1.34)	4.05	(1.89)	5.00	(1.89)	67
drown	591	1.92	(1.48)	6.57	(2.33)	2.86	(1.99)	3
dummy	732	3.38	(1.70)	4.35	(2.25)	3.67	(2.02)	3
dump	733	3.21	(1.87)	4.12	(2.36)	3.83	(1.87)	4
dustpan	555	3.98	(1.68)	3.43	(2.00)	5.45	(1.81)	.
earth	134	7.15	(1.67)	4.24	(2.49)	5.61	(2.30)	150
easy	734	7.10	(1.91)	4.48	(2.82)	7.00	(1.63)	125
easygoing	135	7.20	(1.50)	4.30	(2.52)	5.25	(1.75)	1
eat	136	7.47	(1.73)	5.69	(2.51)	5.60	(2.12)	61
ecstasy	735	7.98	(1.52)	7.38	(1.92)	6.68	(2.08)	6
education	137	6.69	(1.77)	5.74	(2.46)	6.15	(2.35)	214
egg	736	5.29	(1.82)	3.76	(2.39)	4.49	(2.16)	12
elated	138	7.45	(1.77)	6.21	(2.30)	5.53	(2.35)	3
elbow	737	5.12	(0.92)	3.81	(2.14)	4.88	(1.52)	10
elegant	139	7.43	(1.26)	4.53	(2.65)	5.95	(2.09)	14
elevator	738	5.44	(1.18)	4.16	(1.99)	4.32	(1.69)	12
embarrassed	140	3.03	(1.85)	5.87	(2.55)	2.87	(1.99)	8
embattled	141	4.39	(1.63)	5.36	(2.37)	4.81	(1.79)	1
employment	147	6.47	(1.81)	5.28	(2.12)	5.73	(2.08)	47
engaged	143	8.00	(1.38)	6.77	(2.07)	6.49	(2.22)	47
engine	148	5.20	(1.18)	3.98	(2.33)	5.00	(1.77)	50
enjoyment	145	7.80	(1.20)	5.20	(2.72)	6.46	(1.77)	21
ennui	146	5.09	(1.76)	4.40	(2.33)	4.67	(1.80)	.
enraged	149	2.46	(1.65)	7.97	(2.17)	6.33	(2.92)	1
erotic	512	7.43	(1.53)	7.24	(1.97)	6.39	(2.16)	8
errand	150	4.58	(1.74)	3.85	(1.92)	4.78	(1.51)	7
event	740	6.21	(1.63)	5.10	(2.40)	5.52	(1.57)	81
evil	741	3.23	(2.64)	6.39	(2.44)	5.25	(2.60)	72
excellence	151	8.38	(0.96)	5.54	(2.67)	7.28	(2.32)	15
excitement	152	7.50	(2.20)	7.67	(1.91)	6.18	(2.17)	32
excuse	153	4.05	(1.41)	4.48	(2.29)	4.07	(2.10)	27
execution	154	2.37	(2.06)	5.71	(2.74)	4.11	(2.66)	15
exercise	155	7.13	(1.58)	6.84	(2.06)	5.68	(2.44)	58
fabric	742	5.30	(1.20)	4.14	(1.98)	5.03	(1.61)	15
face	556	6.39	(1.60)	5.04	(2.18)	5.67	(1.58)	371
failure	156	1.70	(1.07)	4.95	(2.81)	2.40	(2.18)	89

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
fall	743	4.09	(2.21)	4.70	(2.48)	4.00	(2.15)	147
FALSE	744	3.27	(1.40)	3.43	(2.09)	4.10	(1.56)	29
fame	157	7.93	(1.29)	6.55	(2.46)	6.85	(2.14)	18
family	158	7.65	(1.55)	4.80	(2.71)	6.00	(1.87)	331
famous	745	6.98	(2.07)	5.73	(2.68)	6.32	(2.18)	89
fantasy	746	7.41	(1.90)	5.14	(2.82)	6.43	(2.05)	14
farm	557	5.53	(1.85)	3.90	(1.95)	5.59	(1.81)	125
fascinate	159	7.34	(1.68)	5.83	(2.73)	6.15	(1.89)	3
fat	160	2.28	(1.92)	4.81	(2.80)	4.47	(3.06)	60
father	161	7.08	(2.20)	5.92	(2.60)	5.63	(2.89)	383
fatigued	162	3.28	(1.43)	2.64	(2.19)	3.78	(1.97)	3
fault	747	3.43	(1.38)	4.07	(1.69)	4.02	(1.66)	22
favor	748	6.46	(1.52)	4.54	(1.86)	5.67	(1.76)	78
fear	592	2.76	(2.12)	6.96	(2.17)	3.22	(2.20)	127
fearful	163	2.25	(1.18)	6.33	(2.28)	3.64	(2.18)	13
feeble	164	3.26	(1.47)	4.10	(2.07)	2.71	(1.64)	8
festive	749	7.30	(2.26)	6.58	(2.29)	5.77	(2.34)	2
fever	750	2.76	(1.64)	4.29	(2.31)	3.52	(2.15)	19
field	558	6.20	(1.37)	4.08	(2.41)	5.84	(1.94)	274
fight	751	3.76	(2.63)	7.15	(2.19)	5.27	(2.69)	98
filth	165	2.47	(1.68)	5.12	(2.32)	3.81	(2.06)	2
finger	752	5.29	(1.42)	3.78	(2.42)	5.05	(1.70)	40
fire	166	3.22	(2.06)	7.17	(2.06)	4.49	(2.49)	187
fireworks	513	7.55	(1.50)	6.67	(2.12)	5.51	(1.98)	5
fish	559	6.04	(1.94)	4.00	(2.19)	6.02	(1.68)	35
flabby	167	2.66	(1.87)	4.82	(2.81)	3.31	(1.90)	.
flag	753	6.02	(1.66)	4.60	(2.35)	5.50	(1.66)	16
flirt	754	7.52	(1.19)	6.91	(1.69)	6.24	(2.33)	1
flood	755	3.19	(1.66)	6.00	(2.02)	3.24	(2.14)	19
flower	168	6.64	(1.78)	4.00	(2.44)	4.98	(2.17)	23
foam	756	6.07	(2.03)	5.26	(2.54)	5.24	(1.97)	37
food	514	7.65	(1.37)	5.92	(2.11)	6.18	(2.48)	147
foot	757	5.02	(0.93)	3.27	(1.98)	4.98	(1.42)	70
fork	560	5.29	(0.97)	3.96	(1.94)	5.74	(1.52)	14
foul	169	2.81	(1.52)	4.93	(2.23)	4.51	(1.89)	4
fragrance	170	6.07	(1.97)	4.79	(2.54)	5.14	(1.91)	6
fraud	171	2.67	(1.66)	5.75	(2.45)	3.58	(2.50)	8
free	172	8.26	(1.31)	5.15	(3.04)	6.35	(2.40)	260
freedom	173	7.58	(2.04)	5.52	(2.72)	6.76	(2.29)	128
friend	174	7.74	(1.24)	5.74	(2.57)	6.74	(1.89)	133
friendly	175	8.43	(1.08)	5.11	(2.96)	5.92	(2.42)	61
frigid	758	3.50	(1.85)	4.75	(2.56)	4.27	(1.98)	5
frog	176	5.71	(1.74)	4.54	(2.03)	5.34	(1.96)	1
frustrated	177	2.48	(1.64)	5.61	(2.76)	3.50	(2.12)	10
fun	759	8.37	(1.11)	7.22	(2.01)	6.80	(1.85)	44
funeral	178	1.39	(0.87)	4.94	(3.21)	2.97	(2.55)	33
fungus	179	3.06	(1.75)	4.68	(2.33)	4.06	(1.94)	2
fur	180	4.51	(1.88)	4.18	(2.44)	4.32	(1.97)	13
game	760	6.98	(1.97)	5.89	(2.37)	5.70	(1.65)	123
gangrene	181	2.28	(1.91)	5.70	(2.96)	3.36	(2.34)	.
garbage	182	2.98	(1.96)	5.04	(2.50)	4.24	(2.02)	7
garden	761	6.71	(1.74)	4.39	(2.35)	6.02	(1.71)	60
garment	762	6.07	(1.61)	4.49	(2.50)	5.30	(1.96)	6
garter	534	6.22	(1.59)	5.47	(2.15)	5.82	(1.62)	2
gender	763	5.73	(1.55)	4.38	(2.13)	5.60	(1.84)	2
gentle	183	7.31	(1.30)	3.21	(2.57)	5.10	(2.16)	27
germs	764	2.86	(1.39)	4.49	(2.24)	3.79	(1.59)	1
gift	184	7.77	(2.24)	6.14	(2.76)	5.52	(2.54)	33
girl	185	6.87	(1.64)	4.29	(2.69)	5.80	(2.16)	220
glacier	186	5.50	(1.25)	4.24	(2.29)	4.92	(2.12)	1
glamour	187	6.76	(1.60)	4.68	(2.23)	5.76	(2.49)	5
glass	765	4.75	(1.38)	4.27	(2.07)	5.00	(1.46)	99
gloom	188	1.88	(1.23)	3.83	(2.33)	3.55	(2.07)	14
glory	189	7.55	(1.68)	6.02	(2.71)	6.85	(2.23)	21
god	190	8.15	(1.27)	5.95	(2.84)	5.88	(2.89)	318
gold	191	7.54	(1.63)	5.76	(2.79)	5.85	(2.46)	52
golfer	535	5.61	(1.93)	3.73	(2.26)	5.55	(1.79)	3
good	766	7.47	(1.45)	5.43	(2.85)	6.41	(2.05)	807
gossip	767	3.48	(2.33)	5.74	(2.38)	3.57	(2.26)	13
graduate	192	8.19	(1.13)	7.25	(2.25)	6.94	(2.44)	30
grass	768	6.12	(1.44)	4.14	(2.11)	5.44	(1.36)	53
grateful	193	7.37	(0.97)	4.58	(2.14)	6.18	(1.77)	25
greed	769	3.51	(1.93)	4.71	(2.26)	4.88	(2.03)	3
green	194	6.18	(2.05)	4.28	(2.46)	4.82	(2.05)	116
greet	770	7.00	(1.52)	5.27	(2.31)	5.95	(2.07)	7
grenade	771	3.60	(1.88)	5.70	(2.52)	4.29	(2.50)	3
grief	195	1.69	(1.04)	4.78	(2.84)	3.50	(2.35)	10
grime	772	3.37	(1.34)	3.98	(2.29)	4.47	(1.28)	.

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
grin	773	7.40	(1.87)	5.27	(2.64)	6.00	(1.86)	13
gripe	774	3.14	(1.56)	5.00	(2.19)	4.67	(1.79)	.
guillotine	196	2.48	(2.11)	6.56	(2.54)	4.64	(2.63)	.
guilty	197	2.63	(1.98)	6.04	(2.76)	3.09	(2.22)	29
gun	593	3.47	(2.48)	7.02	(1.84)	3.53	(2.72)	118
gymnast	515	6.35	(1.79)	5.02	(2.20)	5.31	(1.79)	1
habit	775	4.11	(1.77)	3.95	(2.11)	4.30	(1.79)	23
hairdryer	561	4.84	(0.84)	3.71	(1.75)	5.57	(1.27)	.
hairpin	776	5.26	(1.45)	3.27	(2.41)	5.05	(1.32)	1
hamburger	777	6.27	(1.50)	4.55	(2.14)	5.32	(1.21)	6
hammer	198	4.88	(1.16)	4.58	(2.02)	4.75	(1.88)	9
hand	778	5.95	(1.38)	4.40	(2.07)	5.35	(1.49)	431
handicap	779	3.29	(1.69)	3.81	(2.27)	4.00	(2.24)	6
handsome	199	7.93	(1.47)	5.95	(2.73)	5.19	(2.22)	40
haphazard	780	4.02	(1.41)	4.07	(2.18)	4.29	(1.67)	2
happy	200	8.21	(1.82)	6.49	(2.77)	6.63	(2.43)	98
hard	781	5.22	(1.82)	5.12	(2.19)	5.59	(1.63)	202
hardship	782	2.45	(1.61)	4.76	(2.55)	4.22	(2.40)	9
hat	783	5.46	(1.36)	4.10	(2.00)	5.39	(1.43)	56
hate	201	2.12	(1.72)	6.95	(2.56)	5.05	(2.95)	42
hatred	202	1.98	(1.92)	6.66	(2.56)	4.30	(2.76)	20
hawk	536	5.88	(1.62)	4.39	(2.29)	5.50	(1.69)	14
hay	784	5.24	(1.24)	3.95	(2.58)	5.37	(1.64)	19
headache	203	2.02	(1.06)	5.07	(2.74)	3.60	(1.98)	5
headlight	785	5.24	(1.51)	3.81	(2.22)	4.88	(1.47)	.
heal	786	7.09	(1.46)	4.77	(2.23)	5.79	(1.80)	2
health	204	6.81	(1.88)	5.13	(2.35)	5.83	(1.91)	105
heart	787	7.39	(1.53)	6.34	(2.25)	5.49	(2.11)	173
heaven	205	7.30	(2.39)	5.61	(3.20)	6.15	(2.56)	43
hell	788	2.24	(1.62)	5.38	(2.62)	3.24	(2.36)	95
helpless	206	2.20	(1.42)	5.34	(2.52)	2.27	(1.83)	21
heroin	789	4.36	(2.73)	5.11	(2.72)	4.80	(2.54)	2
hide	207	4.32	(1.91)	5.28	(2.51)	3.40	(2.12)	22
highway	562	5.92	(1.72)	5.16	(2.44)	5.66	(1.81)	40
hinder	790	3.81	(1.42)	4.12	(2.01)	4.21	(1.54)	.
history	208	5.24	(2.01)	3.93	(2.29)	4.83	(2.08)	286
hit	594	4.33	(2.35)	5.73	(2.09)	4.88	(2.01)	115
holiday	791	7.55	(2.14)	6.59	(2.73)	6.30	(2.17)	17
home	209	7.91	(1.63)	4.21	(2.94)	5.90	(2.30)	547

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
honest	210	7.70	(1.43)	5.32	(1.92)	6.24	(2.13)	47
honey	792	6.73	(1.70)	4.51	(2.25)	5.44	(1.47)	25
honor	211	7.66	(1.24)	5.90	(1.83)	6.70	(2.04)	66
hooker	793	3.34	(2.31)	4.93	(2.82)	4.73	(2.48)	.
hope	794	7.05	(1.96)	5.44	(2.47)	5.52	(2.20)	178
hopeful	212	7.10	(1.46)	5.78	(2.09)	5.41	(1.92)	12
horror	213	2.76	(2.25)	7.21	(2.14)	4.63	(2.70)	17
horse	214	5.89	(1.55)	3.89	(2.17)	4.67	(1.60)	117
hospital	215	5.04	(2.45)	5.98	(2.54)	4.69	(2.16)	110
hostage	216	2.20	(1.80)	6.76	(2.63)	2.83	(2.32)	2
hostile	217	2.73	(1.50)	6.44	(2.28)	4.85	(2.58)	19
hotel	795	6.00	(1.77)	4.80	(2.53)	5.12	(1.84)	126
house	563	7.26	(1.72)	4.56	(2.41)	6.08	(2.12)	591
hug	218	8.00	(1.55)	5.35	(2.76)	5.79	(2.41)	3
humane	796	6.89	(1.70)	4.50	(1.91)	5.70	(1.91)	5
humble	219	5.86	(1.42)	3.74	(2.33)	4.76	(2.25)	18
humiliate	797	2.24	(1.34)	6.14	(2.42)	2.60	(1.94)	.
humor	220	8.56	(0.81)	5.50	(2.91)	6.08	(2.14)	47
hungry	221	3.58	(2.01)	5.13	(2.44)	4.68	(2.05)	23
hurricane	798	3.34	(2.12)	6.83	(2.06)	3.07	(2.18)	8
hurt	222	1.90	(1.26)	5.85	(2.49)	3.33	(2.22)	37
hydrant	564	5.02	(0.93)	3.71	(1.75)	5.53	(1.30)	.
icebox	799	4.95	(1.00)	4.17	(2.11)	5.05	(1.05)	3
idea	800	7.00	(1.34)	5.86	(1.81)	6.26	(2.00)	195
identity	801	6.57	(1.99)	4.95	(2.24)	6.40	(1.89)	55
idiot	223	3.16	(1.91)	4.21	(2.47)	3.18	(2.13)	2
idol	802	6.12	(1.86)	4.95	(2.14)	5.37	(2.17)	7
ignorance	803	3.07	(2.25)	4.39	(2.49)	4.41	(2.38)	16
illness	804	2.48	(1.40)	4.71	(2.24)	3.21	(1.85)	20
imagine	805	7.32	(1.52)	5.98	(2.14)	7.07	(1.99)	61
immature	806	3.39	(1.70)	4.15	(1.96)	4.85	(2.20)	7
immoral	807	3.50	(2.16)	4.98	(2.48)	4.66	(2.33)	5
impair	808	3.18	(1.86)	4.04	(2.14)	4.09	(2.18)	4
impotent	224	2.81	(1.92)	4.57	(2.59)	3.43	(2.43)	2
impressed	225	7.33	(1.84)	5.42	(2.65)	5.51	(2.21)	30
improve	226	7.65	(1.16)	5.69	(2.15)	6.08	(2.25)	39
incentive	809	7.00	(1.72)	5.69	(2.45)	5.93	(2.02)	12
indifferent	810	4.61	(1.28)	3.18	(1.85)	4.84	(1.67)	11
industry	227	5.30	(1.61)	4.47	(2.43)	4.91	(2.04)	171

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
infant	811	6.95	(2.08)	5.05	(2.66)	5.67	(2.48)	11
infatuation	516	6.73	(2.08)	7.02	(1.87)	4.90	(2.28)	4
infection	228	1.66	(1.34)	5.03	(2.77)	3.61	(2.64)	8
inferior	812	3.07	(1.57)	3.83	(2.05)	2.78	(2.08)	7
inhabitant	813	5.05	(1.34)	3.95	(1.97)	5.37	(1.43)	.
injury	595	2.49	(1.76)	5.69	(2.06)	3.57	(1.62)	27
ink	229	5.05	(0.81)	3.84	(1.88)	4.61	(2.13)	7
innocent	814	6.51	(1.34)	4.21	(1.99)	5.28	(2.08)	37
insane	815	2.85	(1.94)	5.83	(2.45)	4.12	(2.23)	13
insect	816	4.07	(2.16)	4.07	(2.46)	4.56	(2.47)	14
insecure	230	2.36	(1.33)	5.56	(2.34)	2.33	(1.95)	3
insolent	231	4.35	(1.76)	5.38	(2.37)	4.50	(2.06)	2
inspire	232	6.97	(1.91)	5.00	(2.53)	6.34	(2.11)	3
inspired	233	7.15	(1.85)	6.02	(2.67)	6.67	(2.31)	25
insult	817	2.29	(1.33)	6.00	(2.46)	3.62	(2.05)	7
intellect	818	6.82	(1.96)	4.75	(2.50)	6.30	(1.98)	5
intercourse	819	7.36	(1.57)	7.00	(2.07)	6.40	(1.78)	9
interest	234	6.97	(1.53)	5.66	(2.26)	5.88	(1.78)	330
intimate	821	7.61	(1.51)	6.98	(2.21)	5.86	(2.29)	21
intruder	822	2.77	(2.32)	6.86	(2.41)	4.00	(2.68)	1
invader	823	3.05	(2.01)	5.50	(2.40)	4.00	(2.60)	1
invest	824	5.93	(2.10)	5.12	(2.42)	5.88	(1.95)	3
iron	565	4.90	(1.02)	3.76	(2.06)	5.10	(1.27)	43
irritate	235	3.11	(1.67)	5.76	(2.15)	5.03	(2.05)	.
item	825	5.26	(0.86)	3.24	(2.08)	5.26	(1.67)	54
jail	236	1.95	(1.27)	5.49	(2.67)	3.81	(2.71)	21
jealousy	237	2.51	(1.83)	6.36	(2.66)	3.80	(2.41)	4
jelly	238	5.66	(1.44)	3.70	(2.29)	4.53	(1.77)	3
jewel	239	7.00	(1.72)	5.38	(2.54)	5.59	(2.19)	1
joke	826	8.10	(1.36)	6.74	(1.84)	6.15	(1.86)	22
jolly	827	7.41	(1.92)	5.57	(2.80)	6.39	(1.72)	4
journal	828	5.14	(1.49)	4.05	(1.96)	5.26	(1.42)	42
joy	240	8.60	(0.71)	7.22	(2.13)	6.28	(2.15)	40
joyful	241	8.22	(1.22)	5.98	(2.54)	6.60	(1.80)	1
jug	829	5.24	(1.65)	3.88	(2.15)	5.05	(1.62)	6
justice	242	7.78	(1.35)	5.47	(2.54)	6.47	(2.26)	114
kerchief	830	5.11	(1.33)	3.43	(2.08)	5.25	(1.28)	1
kerosene	243	4.80	(1.59)	4.34	(2.51)	4.63	(1.99)	6
ketchup	831	5.60	(1.35)	4.09	(2.08)	5.29	(1.81)	1
kettle	832	5.22	(0.91)	3.22	(2.23)	5.00	(1.40)	3
key	833	5.68	(1.62)	3.70	(2.18)	4.98	(2.04)	88
kick	834	4.31	(2.18)	4.90	(2.35)	5.50	(1.93)	16
kids	835	6.91	(1.99)	5.27	(2.36)	5.07	(2.03)	32
killer	244	1.89	(1.39)	7.86	(1.89)	4.54	(3.11)	21
kind	245	7.59	(1.67)	4.46	(2.55)	5.95	(1.93)	313
kindness	246	7.82	(1.39)	4.30	(2.62)	5.67	(2.63)	5
king	247	7.26	(1.67)	5.51	(2.77)	7.38	(2.10)	88
kiss	248	8.26	(1.54)	7.32	(2.03)	6.93	(2.28)	17
kitten	517	6.86	(2.13)	5.08	(2.45)	6.86	(2.01)	5
knife	596	3.62	(2.18)	5.80	(2.00)	4.12	(2.18)	76
knot	836	4.64	(1.36)	4.07	(2.15)	4.67	(1.65)	8
knowledge	249	7.58	(1.32)	5.92	(2.32)	6.78	(2.41)	145
lake	250	6.82	(1.54)	3.95	(2.44)	4.90	(2.10)	54
lamb	837	5.89	(1.73)	3.36	(2.18)	4.91	(1.96)	7
lamp	838	5.41	(1.00)	3.80	(2.12)	5.27	(1.61)	18
lantern	839	5.57	(1.19)	4.05	(2.28)	5.07	(1.82)	13
laughter	251	8.45	(1.08)	6.75	(2.50)	6.45	(2.45)	22
lavish	840	6.21	(2.03)	4.93	(2.40)	5.64	(1.61)	3
lawn	841	5.24	(0.86)	4.00	(1.79)	5.37	(1.11)	15
lawsuit	842	3.37	(2.00)	4.93	(2.44)	3.92	(2.02)	1
lazy	843	4.38	(2.02)	2.65	(2.06)	4.07	(1.93)	9
leader	844	7.63	(1.59)	6.27	(2.18)	7.88	(1.60)	74
learn	252	7.15	(1.49)	5.39	(2.22)	6.34	(2.17)	84
legend	845	6.39	(1.34)	4.88	(1.76)	5.54	(1.64)	26
leisurely	253	6.88	(1.81)	3.80	(2.38)	5.15	(1.90)	5
leprosy	254	2.09	(1.40)	6.29	(2.23)	4.00	(2.30)	1
lesbian	597	4.67	(2.45)	5.12	(2.27)	5.35	(2.20)	.
letter	846	6.61	(1.59)	4.90	(2.37)	5.73	(1.48)	145
liberty	255	7.98	(1.22)	5.60	(2.65)	6.29	(2.44)	46
lice	256	2.31	(1.78)	5.00	(2.26)	3.95	(2.29)	2
lie	257	2.79	(1.92)	5.96	(2.63)	3.30	(2.42)	59
life	258	7.27	(1.88)	6.02	(2.62)	5.72	(2.51)	715
lightbulb	566	5.61	(1.28)	4.10	(2.02)	5.82	(1.56)	.
lighthouse	847	5.89	(2.08)	4.41	(2.44)	5.25	(2.02)	.
lightning	598	4.57	(2.66)	6.61	(1.77)	3.67	(2.19)	14
limber	848	5.68	(1.49)	4.57	(2.26)	5.34	(1.84)	2
lion	518	5.57	(1.99)	6.20	(2.16)	4.12	(2.33)	17
listless	259	4.12	(1.73)	4.10	(2.31)	4.14	(1.73)	1

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
lively	849	7.20	(1.97)	5.53	(2.90)	6.09	(1.95)	26
locker	850	5.19	(1.31)	3.38	(2.13)	5.36	(1.87)	9
loneliness	260	1.61	(1.02)	4.56	(2.97)	2.51	(2.27)	9
lonely	261	2.17	(1.76)	4.51	(2.68)	2.95	(2.12)	25
loser	851	2.25	(1.48)	4.95	(2.57)	3.02	(2.17)	1
lost	852	2.82	(1.83)	5.82	(2.62)	2.86	(1.64)	173
lottery	853	6.57	(2.04)	5.36	(2.45)	4.81	(2.11)	1
louse	262	2.81	(1.92)	4.98	(2.03)	3.57	(2.26)	3
love	263	8.72	(0.70)	6.44	(3.35)	7.11	(2.56)	232
loved	264	8.64	(0.71)	6.38	(2.68)	6.62	(2.53)	56
loyal	265	7.55	(1.90)	5.16	(2.42)	6.91	(2.23)	18
lucky	266	8.17	(1.06)	6.53	(2.34)	6.05	(2.25)	21
lump	854	4.16	(2.34)	4.80	(2.82)	4.32	(2.18)	7
luscious	267	7.50	(1.08)	5.34	(2.51)	5.68	(1.84)	2
lust	519	7.12	(1.62)	6.88	(1.85)	5.49	(2.27)	5
luxury	268	7.88	(1.49)	4.75	(2.91)	6.40	(2.45)	21
machine	855	5.09	(1.67)	3.82	(2.40)	5.23	(2.06)	103
mad	856	2.44	(1.72)	6.76	(2.26)	5.86	(2.20)	39
madman	857	3.91	(2.49)	5.56	(2.78)	4.79	(2.55)	2
maggot	269	2.06	(1.47)	5.28	(2.96)	4.03	(2.09)	2
magical	858	7.46	(1.64)	5.95	(2.36)	5.73	(2.19)	12
mail	859	6.88	(1.74)	5.63	(2.36)	5.67	(1.79)	47
malaria	860	2.40	(1.38)	4.40	(2.54)	3.22	(1.90)	3
malice	270	2.69	(1.84)	5.86	(2.75)	4.74	(2.72)	2
man	537	6.73	(1.70)	5.24	(2.31)	5.53	(2.23)	1207
mangle	861	3.90	(2.01)	5.44	(2.10)	4.61	(1.84)	.
maniac	862	3.76	(2.00)	5.39	(2.46)	4.22	(2.07)	4
manner	863	5.64	(1.34)	4.56	(1.78)	5.05	(1.83)	124
mantel	864	4.93	(1.40)	3.27	(2.23)	4.95	(1.61)	3
manure	865	3.10	(1.74)	4.17	(2.09)	4.67	(1.36)	6
market	866	5.66	(1.02)	4.12	(1.83)	5.27	(1.40)	155
massacre	867	2.28	(1.74)	5.33	(2.63)	3.50	(2.26)	1
masterful	271	7.09	(1.78)	5.20	(2.85)	7.18	(2.56)	2
masturbate	599	5.45	(2.02)	5.67	(2.18)	5.63	(2.25)	.
material	868	5.26	(1.29)	4.05	(2.34)	5.12	(1.45)	174
measles	272	2.74	(1.97)	5.06	(2.44)	4.13	(2.16)	2
medicine	869	5.67	(2.06)	4.40	(2.36)	4.70	(1.91)	30
meek	273	3.87	(1.69)	3.80	(2.13)	3.67	(2.23)	10
melody	870	7.07	(1.79)	4.98	(2.52)	5.46	(1.78)	21
memories	871	7.48	(1.61)	6.10	(2.10)	5.88	(1.92)	15
memory	274	6.62	(1.50)	5.42	(2.25)	5.11	(2.12)	76
menace	275	2.88	(1.64)	5.52	(2.45)	4.98	(2.25)	9
merry	872	7.90	(1.49)	5.90	(2.42)	6.64	(1.66)	8
messy	873	3.15	(1.73)	3.34	(2.37)	4.75	(2.15)	3
metal	874	4.95	(1.17)	3.79	(1.96)	5.38	(1.40)	61
method	875	5.56	(1.76)	3.85	(2.58)	5.67	(1.58)	142
mighty	276	6.54	(2.19)	5.61	(2.38)	7.23	(2.11)	29
mildew	277	3.17	(1.36)	4.08	(1.79)	4.40	(1.79)	1
milk	876	5.95	(2.16)	3.68	(2.57)	5.83	(1.50)	49
millionaire	278	8.03	(1.42)	6.14	(2.70)	6.97	(2.40)	2
mind	877	6.68	(1.84)	5.00	(2.68)	6.37	(2.19)	325
miracle	279	8.60	(0.71)	7.65	(1.67)	5.35	(2.58)	16
mischief	878	5.57	(2.05)	5.76	(1.95)	5.56	(1.88)	5
misery	879	1.93	(1.60)	5.17	(2.69)	2.55	(1.45)	15
mistake	880	2.86	(1.79)	5.18	(2.42)	3.86	(2.42)	34
mobility	881	6.83	(1.79)	5.00	(2.18)	6.43	(1.48)	8
modest	280	5.76	(1.28)	3.98	(2.24)	4.96	(2.16)	29
mold	882	3.55	(1.70)	4.07	(1.98)	4.33	(1.83)	45
moment	281	5.76	(1.65)	3.83	(2.29)	4.81	(1.92)	246
money	282	7.59	(1.40)	5.70	(2.66)	6.25	(2.33)	265
month	283	5.15	(1.09)	4.03	(1.77)	4.85	(1.14)	130
moody	883	3.20	(1.58)	4.18	(2.38)	4.39	(1.71)	5
moral	884	6.20	(1.85)	4.49	(2.28)	5.90	(2.20)	142
morbid	284	2.87	(2.14)	5.06	(2.68)	4.34	(2.50)	1
morgue	285	1.92	(1.32)	4.84	(2.96)	3.61	(1.94)	1
mosquito	885	2.80	(1.91)	4.78	(2.72)	4.51	(2.15)	1
mother	286	8.39	(1.15)	6.13	(2.71)	5.74	(2.37)	216
mountain	287	6.59	(1.66)	5.49	(2.43)	5.46	(2.36)	33
movie	288	6.86	(1.81)	4.93	(2.54)	5.00	(1.79)	29
mucus	886	3.34	(2.29)	3.41	(2.17)	4.80	(1.83)	2
muddy	887	4.44	(2.07)	4.13	(2.13)	4.73	(1.77)	10
muffin	888	6.57	(2.04)	4.76	(2.42)	5.51	(1.63)	.
murderer	289	1.53	(0.96)	7.47	(2.18)	3.77	(3.06)	19
muscular	290	6.82	(1.63)	5.47	(2.20)	6.58	(2.28)	16
museum	889	5.54	(1.86)	3.60	(2.13)	5.32	(1.68)	32
mushroom	567	5.78	(2.22)	4.72	(2.33)	5.52	(2.10)	2
music	291	8.13	(1.09)	5.32	(3.19)	6.39	(2.44)	216
mutation	890	3.91	(2.44)	4.84	(2.52)	4.07	(2.10)	.

Affective Norms for English Words. **All Subjects** *Table 1*

Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
mutilate	292	1.82	(1.45)	6.41	(2.94)	3.41	(2.71)	3
mystic	891	6.00	(2.21)	4.84	(2.57)	5.52	(1.93)	3
naked	892	6.34	(2.42)	5.80	(2.80)	6.00	(2.05)	32
name	893	5.55	(2.24)	4.25	(2.47)	5.16	(2.08)	294
narcotic	894	4.29	(2.30)	4.93	(2.57)	4.44	(2.43)	2
nasty	895	3.58	(2.38)	4.89	(2.50)	5.00	(2.17)	5
natural	896	6.59	(1.57)	4.09	(2.37)	5.57	(1.69)	156
nature	293	7.65	(1.37)	4.37	(2.51)	4.95	(2.72)	191
nectar	294	6.90	(1.53)	3.89	(2.48)	4.54	(2.06)	3
needle	897	3.82	(1.73)	5.36	(2.89)	3.95	(2.17)	15
neglect	898	2.63	(1.64)	4.83	(2.31)	3.85	(2.29)	12
nervous	899	3.29	(1.47)	6.59	(2.07)	3.56	(1.73)	24
neurotic	900	4.45	(2.23)	5.13	(2.76)	4.41	(2.05)	10
news	901	5.30	(1.67)	5.17	(2.11)	4.60	(1.88)	102
nice	902	6.55	(2.44)	4.38	(2.69)	5.58	(2.20)	75
nightmare	295	1.91	(1.54)	7.59	(2.23)	3.68	(2.76)	9
nipple	903	6.27	(1.81)	5.56	(2.55)	5.57	(2.00)	.
noisy	904	5.02	(2.02)	6.38	(1.78)	4.93	(1.76)	6
nonchalant	296	4.74	(1.11)	3.12	(1.93)	4.31	(1.54)	1
nonsense	905	4.61	(1.63)	4.17	(2.02)	4.90	(1.55)	13
noose	906	3.76	(1.64)	4.39	(2.08)	4.17	(1.92)	3
nourish	907	6.46	(1.69)	4.29	(2.51)	5.80	(1.62)	.
nude	520	6.82	(1.63)	6.41	(2.09)	5.96	(2.29)	20
nuisance	908	3.27	(1.86)	4.49	(2.69)	4.36	(1.73)	5
nun	909	4.93	(1.89)	2.93	(1.80)	4.93	(1.69)	2
nurse	538	6.08	(2.08)	4.84	(2.04)	4.84	(2.20)	17
nursery	910	5.73	(2.30)	4.04	(2.74)	5.18	(2.23)	13
obesity	911	2.73	(1.85)	3.87	(2.82)	3.74	(2.45)	5
obey	912	4.52	(1.88)	4.23	(1.72)	4.26	(2.40)	8
obnoxious	913	3.50	(2.18)	4.74	(2.42)	5.39	(2.20)	5
obscene	914	4.23	(2.30)	5.04	(2.30)	4.48	(1.91)	2
obsession	915	4.52	(2.13)	6.41	(2.13)	4.77	(2.38)	5
ocean	297	7.12	(1.72)	4.95	(2.79)	5.53	(2.75)	34
odd	916	4.82	(2.04)	4.27	(2.46)	4.77	(1.89)	44
offend	917	2.76	(1.50)	5.56	(2.06)	3.73	(2.03)	4
office	568	5.24	(1.59)	4.08	(1.92)	5.59	(1.89)	255
opinion	298	6.28	(1.45)	4.89	(2.46)	5.53	(1.93)	96
optimism	918	6.95	(2.24)	5.34	(2.58)	6.61	(2.06)	15
option	919	6.49	(1.31)	4.74	(2.23)	6.34	(1.80)	5
orchestra	299	6.02	(1.89)	3.52	(2.29)	5.17	(2.14)	60
orgasm	920	8.32	(1.31)	8.10	(1.45)	6.83	(2.18)	7
outdoors	521	7.47	(1.80)	5.92	(2.55)	6.27	(2.24)	6
outrage	921	3.52	(2.12)	6.83	(2.26)	5.26	(2.72)	4
outstanding	922	7.75	(1.75)	6.24	(2.59)	6.40	(2.29)	37
overcast	923	3.65	(1.61)	3.46	(1.92)	4.20	(1.79)	9
overwhelmed	300	4.19	(2.61)	7.00	(2.37)	3.89	(2.58)	4
owl	522	5.80	(1.31)	3.98	(1.87)	5.82	(1.62)	2
pain	301	2.13	(1.81)	6.50	(2.49)	3.71	(2.53)	88
paint	924	5.62	(1.72)	4.10	(2.36)	5.75	(1.71)	37
palace	302	7.19	(1.78)	5.10	(2.75)	5.69	(2.17)	38
pamphlet	925	4.79	(1.05)	3.62	(2.02)	4.63	(1.48)	3
pancakes	523	6.08	(1.83)	4.06	(2.13)	5.76	(1.61)	.
panic	601	3.12	(1.84)	7.02	(2.02)	3.20	(1.67)	22
paper	303	5.20	(1.21)	2.50	(1.85)	4.47	(1.67)	157
paradise	304	8.72	(0.60)	5.12	(3.38)	6.03	(2.79)	12
paralysis	926	1.98	(1.44)	4.73	(2.83)	2.56	(1.82)	6
part	927	5.11	(1.78)	3.82	(2.24)	4.75	(1.59)	500
party	305	7.86	(1.83)	6.69	(2.84)	5.83	(2.46)	216
passage	928	5.28	(1.44)	4.36	(2.13)	5.02	(1.62)	49
passion	306	8.03	(1.27)	7.26	(2.57)	6.13	(2.24)	28
pasta	524	6.69	(1.64)	4.94	(2.04)	5.80	(1.47)	.
patent	307	5.29	(1.08)	3.50	(1.84)	4.90	(1.79)	35
patient	929	5.29	(1.89)	4.21	(2.37)	4.90	(2.31)	86
patriot	930	6.71	(1.69)	5.17	(2.53)	5.90	(1.54)	10
peace	308	7.72	(1.75)	2.95	(2.55)	5.45	(2.84)	198
penalty	931	2.83	(1.56)	5.10	(2.31)	3.95	(1.97)	14
pencil	309	5.22	(0.68)	3.14	(1.90)	4.78	(1.73)	34
penis	932	5.90	(1.72)	5.54	(2.63)	5.92	(2.54)	.
penthouse	933	6.81	(1.64)	5.52	(2.49)	6.52	(1.82)	1
people	525	7.33	(1.70)	5.94	(2.09)	6.14	(2.02)	847
perfection	310	7.25	(2.05)	5.95	(2.73)	6.71	(2.26)	11
perfume	934	6.76	(1.48)	5.05	(2.36)	5.93	(1.69)	10
person	311	6.32	(1.74)	4.19	(2.45)	5.35	(2.02)	175
pervert	312	2.79	(2.12)	6.26	(2.61)	4.72	(2.83)	1
pest	313	3.13	(1.82)	5.62	(2.15)	5.29	(2.13)	4
pet	935	6.79	(2.32)	5.10	(2.59)	5.85	(2.28)	8
phase	936	5.17	(0.79)	3.98	(1.82)	4.65	(1.72)	72
pie	314	6.41	(1.89)	4.20	(2.40)	5.35	(1.78)	14

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
pig	937	5.07	(1.97)	4.20	(2.42)	5.34	(1.88)	8
pillow	315	7.92	(1.40)	2.97	(2.52)	4.56	(2.17)	8
pinch	938	3.83	(1.70)	4.59	(2.10)	4.76	(1.73)	6
pistol	939	4.20	(2.58)	6.15	(2.19)	5.05	(2.77)	27
pity	940	3.37	(1.57)	3.72	(2.02)	4.12	(1.82)	14
pizza	526	6.65	(2.23)	5.24	(2.09)	5.69	(1.90)	3
plain	941	4.39	(1.46)	3.52	(2.05)	4.71	(1.68)	48
plane	539	6.43	(1.98)	6.14	(2.39)	4.78	(2.19)	114
plant	316	5.98	(1.83)	3.62	(2.25)	4.71	(2.12)	125
pleasure	317	8.28	(0.92)	5.74	(2.81)	6.15	(2.31)	62
poetry	318	5.86	(1.91)	4.00	(2.85)	5.31	(1.81)	88
poison	319	1.98	(1.44)	6.05	(2.82)	3.10	(2.44)	10
politeness	320	7.18	(1.50)	3.74	(2.37)	5.74	(1.70)	5
pollute	321	1.85	(1.11)	6.08	(2.42)	4.92	(2.51)	1
poster	942	5.34	(1.75)	3.93	(2.56)	4.91	(1.87)	4
poverty	322	1.67	(0.90)	4.87	(2.66)	3.21	(2.21)	20
power	323	6.54	(2.21)	6.67	(1.87)	7.28	(2.35)	342
powerful	324	6.84	(1.80)	5.83	(2.69)	7.19	(2.52)	63
prairie	325	5.75	(1.43)	3.41	(2.17)	4.62	(2.13)	21
present	943	6.95	(1.85)	5.12	(2.39)	5.83	(1.78)	377
pressure	944	3.38	(1.61)	6.07	(2.26)	3.45	(2.07)	185
prestige	945	7.26	(1.90)	5.86	(2.08)	6.90	(1.96)	29
pretty	326	7.75	(1.26)	6.03	(2.22)	5.50	(1.97)	107
prick	946	3.98	(1.73)	4.70	(2.59)	4.47	(1.88)	2
pride	327	7.00	(2.11)	5.83	(2.48)	7.06	(2.15)	42
priest	328	6.42	(2.00)	4.41	(2.71)	4.88	(2.07)	16
prison	329	2.05	(1.34)	5.70	(2.56)	4.20	(2.58)	42
privacy	330	5.88	(1.50)	4.12	(1.83)	5.66	(1.78)	12
profit	331	7.63	(1.30)	6.68	(1.78)	5.85	(2.47)	28
progress	947	7.73	(1.34)	6.02	(2.58)	6.76	(2.05)	120
promotion	332	8.20	(1.15)	6.44	(2.58)	6.79	(2.28)	26
protected	333	7.29	(1.79)	4.09	(2.77)	5.80	(2.54)	31
proud	334	8.03	(1.56)	5.56	(3.01)	6.74	(2.73)	50
pungent	948	3.95	(2.09)	4.24	(2.17)	4.78	(1.52)	4
punishment	335	2.22	(1.41)	5.93	(2.40)	3.50	(2.43)	21
puppy	336	7.56	(1.90)	5.85	(2.78)	5.51	(2.39)	2
pus	602	2.86	(1.91)	4.82	(2.06)	4.35	(1.82)	.
putrid	337	2.38	(1.71)	5.74	(2.26)	4.89	(2.09)	.
python	949	4.05	(2.48)	6.18	(2.25)	4.52	(2.56)	14
quality	950	6.25	(1.59)	4.48	(2.12)	5.64	(1.59)	114
quarrel	338	2.93	(2.06)	6.29	(2.56)	4.02	(2.16)	20
quart	951	5.39	(2.01)	3.59	(2.51)	5.20	(1.86)	3
queen	952	6.44	(1.43)	4.76	(2.18)	5.49	(2.12)	41
quick	953	6.64	(1.61)	6.57	(1.78)	6.57	(1.91)	68
quiet	339	5.58	(1.83)	2.82	(2.13)	4.42	(2.30)	76
rabbit	527	6.57	(1.92)	4.02	(2.19)	6.08	(1.72)	11
rabies	340	1.77	(0.97)	6.10	(2.62)	3.85	(2.34)	1
radiant	954	6.73	(2.17)	5.39	(2.82)	5.61	(2.17)	8
radiator	955	4.67	(1.05)	4.02	(1.94)	4.81	(1.38)	4
radio	341	6.73	(1.47)	4.78	(2.82)	5.28	(1.85)	120
rage	342	2.41	(1.86)	8.17	(1.40)	5.68	(3.01)	16
rain	569	5.08	(2.51)	3.65	(2.35)	4.78	(1.68)	70
rainbow	343	8.14	(1.23)	4.64	(2.88)	4.72	(2.37)	4
rancid	956	4.34	(2.28)	5.04	(2.27)	4.59	(1.86)	.
rape	344	1.25	(0.91)	6.81	(3.17)	2.97	(2.94)	5
rat	345	3.02	(1.66)	4.95	(2.36)	4.55	(2.14)	6
rattle	346	5.03	(1.23)	4.36	(2.18)	4.17	(1.56)	5
razor	957	4.81	(2.16)	5.36	(2.44)	4.91	(1.95)	15
red	570	6.41	(1.61)	5.29	(2.04)	5.78	(1.59)	197
refreshment	347	7.44	(1.29)	4.45	(2.70)	5.00	(1.92)	2
regretful	348	2.28	(1.42)	5.74	(2.32)	3.43	(2.52)	1
rejected	349	1.50	(1.09)	6.37	(2.56)	2.72	(2.58)	33
relaxed	350	7.00	(1.77)	2.39	(2.13)	5.55	(1.90)	14
repentant	351	5.53	(1.86)	4.69	(1.98)	5.42	(2.06)	1
reptile	958	4.77	(2.00)	5.18	(2.19)	4.77	(2.02)	.
rescue	352	7.70	(1.24)	6.53	(2.56)	6.45	(2.29)	15
resent	959	3.76	(1.90)	4.47	(2.12)	4.46	(2.09)	8
reserved	353	4.88	(1.83)	3.27	(2.05)	4.30	(1.93)	27
respect	354	7.64	(1.29)	5.19	(2.39)	6.89	(2.11)	125
respectful	355	7.22	(1.27)	4.60	(2.67)	5.67	(2.38)	4
restaurant	960	6.76	(1.85)	5.41	(2.55)	5.73	(1.41)	41
reunion	961	6.48	(2.45)	6.34	(2.35)	5.64	(1.95)	11
reverent	356	5.35	(1.21)	4.00	(1.60)	4.67	(1.68)	3
revolt	357	4.13	(1.78)	6.56	(2.34)	6.18	(2.11)	8
revolver	962	4.02	(2.44)	5.55	(2.39)	4.39	(2.47)	14
reward	358	7.53	(1.67)	4.95	(2.62)	6.00	(2.14)	15
riches	359	7.70	(1.95)	6.17	(2.70)	6.74	(2.43)	2
ridicule	360	3.13	(2.24)	5.83	(2.73)	3.87	(2.70)	5

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
rifle	603	4.02	(2.76)	6.35	(2.04)	4.16	(2.71)	63
rigid	963	3.66	(2.12)	4.66	(2.47)	4.61	(2.04)	24
riot	361	2.96	(1.93)	6.39	(2.63)	4.18	(2.47)	7
river	362	6.85	(1.69)	4.51	(2.42)	5.10	(1.86)	165
roach	363	2.35	(1.70)	6.64	(2.64)	4.82	(2.94)	2
robber	964	2.61	(1.69)	5.62	(2.72)	3.62	(2.38)	2
rock	965	5.56	(1.38)	4.52	(2.37)	5.15	(2.01)	75
rollercoaster	528	8.02	(1.63)	8.06	(1.71)	5.10	(2.76)	.
romantic	364	8.32	(1.00)	7.59	(2.07)	6.08	(2.29)	32
rotten	365	2.26	(1.37)	4.53	(2.38)	4.32	(2.09)	2
rough	966	4.74	(2.00)	5.33	(2.04)	4.81	(1.70)	41
rude	366	2.50	(2.11)	6.31	(2.47)	4.91	(2.49)	6
runner	571	5.67	(1.91)	4.76	(2.40)	5.47	(1.84)	1
rusty	367	3.86	(1.47)	3.77	(2.16)	4.53	(1.62)	8
sad	368	1.61	(0.95)	4.13	(2.38)	3.45	(2.18)	35
safe	967	7.07	(1.90)	3.86	(2.72)	5.81	(2.06)	58
sailboat	529	7.25	(1.71)	4.88	(2.73)	5.86	(1.71)	1
saint	968	6.49	(1.70)	4.49	(1.90)	5.37	(2.11)	16
salad	369	5.74	(1.62)	3.81	(2.29)	5.47	(1.68)	9
salute	370	5.92	(1.57)	5.31	(2.23)	5.46	(2.05)	3
sapphire	371	7.00	(1.88)	5.00	(2.72)	5.55	(2.24)	.
satisfied	372	7.94	(1.19)	4.94	(2.63)	6.14	(2.37)	36
save	969	6.45	(1.93)	4.95	(2.19)	6.00	(1.79)	62
savior	373	7.73	(1.56)	5.80	(3.01)	6.64	(2.18)	6
scalding	970	2.82	(2.12)	5.95	(2.55)	3.82	(2.30)	1
scandal	971	3.32	(1.81)	5.12	(2.22)	4.34	(1.73)	8
scapegoat	972	3.67	(1.65)	4.53	(2.13)	3.52	(1.70)	1
scar	973	3.38	(1.70)	4.79	(2.11)	3.88	(1.71)	10
scared	604	2.78	(1.99)	6.82	(2.03)	2.94	(2.19)	21
scholar	374	7.26	(1.42)	5.12	(2.46)	6.59	(2.02)	15
scissors	974	5.05	(0.96)	4.47	(1.76)	5.16	(1.84)	1
scorching	975	3.76	(1.83)	5.00	(2.74)	4.10	(2.01)	.
scorn	375	2.84	(2.07)	5.48	(2.52)	3.93	(2.64)	4
scornful	376	3.02	(2.03)	5.04	(2.56)	4.59	(2.18)	5
scorpion	976	3.69	(2.63)	5.38	(3.08)	3.98	(2.44)	.
scream	605	3.88	(2.07)	7.04	(1.96)	4.75	(2.21)	13
scum	377	2.43	(1.56)	4.88	(2.36)	4.26	(1.99)	.
scurvy	378	3.19	(2.00)	4.71	(2.72)	4.48	(2.48)	1
seasick	379	2.05	(1.20)	5.80	(2.88)	3.41	(2.39)	.
seat	380	4.95	(0.98)	2.95	(1.72)	4.84	(1.88)	54
secure	381	7.57	(1.76)	3.14	(2.47)	5.93	(2.57)	30
selfish	382	2.42	(1.62)	5.50	(2.62)	4.64	(2.31)	8
sentiment	977	5.98	(1.71)	4.41	(2.30)	5.09	(1.46)	23
serious	383	5.08	(1.59)	4.00	(1.87)	5.12	(1.65)	116
severe	978	3.20	(1.74)	5.26	(2.36)	3.83	(1.91)	39
sex	384	8.05	(1.53)	7.36	(1.91)	5.75	(2.25)	84
sexy	530	8.02	(1.12)	7.36	(1.91)	6.82	(2.13)	2
shadow	385	4.35	(1.23)	4.30	(2.26)	4.19	(1.82)	36
shamed	386	2.50	(1.34)	4.88	(2.27)	2.98	(1.94)	1
shark	606	3.65	(2.47)	7.16	(1.96)	2.63	(2.16)	.
sheltered	387	5.75	(1.92)	4.28	(1.77)	3.76	(1.91)	4
ship	388	5.55	(1.40)	4.38	(2.29)	5.12	(2.31)	83
shotgun	979	4.37	(2.75)	6.27	(1.94)	5.29	(2.67)	8
shriek	980	3.93	(2.22)	5.36	(2.91)	4.30	(1.86)	5
shy	389	4.64	(1.83)	3.77	(2.29)	3.44	(1.96)	13
sick	607	1.90	(1.14)	4.29	(2.45)	3.04	(1.65)	51
sickness	390	2.25	(1.71)	5.61	(2.67)	3.84	(2.50)	6
silk	391	6.90	(1.27)	3.71	(2.51)	4.81	(1.93)	12
silly	981	7.41	(1.80)	5.88	(2.38)	6.00	(2.09)	15
sin	392	2.80	(1.67)	5.78	(2.21)	3.62	(2.29)	53
sinful	393	2.93	(2.15)	6.29	(2.43)	4.24	(2.73)	3
sissy	394	3.14	(1.96)	5.17	(2.57)	3.58	(2.74)	.
skeptical	395	4.52	(1.63)	4.91	(1.92)	4.50	(1.61)	7
skijump	531	7.06	(1.73)	7.06	(2.10)	4.90	(2.32)	.
skull	608	4.27	(1.83)	4.75	(1.85)	4.86	(1.62)	3
sky	572	7.37	(1.40)	4.27	(2.17)	5.16	(2.00)	58
skyscraper	573	5.88	(1.87)	5.71	(2.17)	4.33	(2.36)	2
slap	396	2.95	(1.79)	6.46	(2.58)	4.21	(2.29)	2
slaughter	397	1.64	(1.18)	6.77	(2.42)	3.82	(2.75)	10
slave	398	1.84	(1.13)	6.21	(2.93)	3.29	(2.76)	30
sleep	399	7.20	(1.77)	2.80	(2.66)	5.41	(2.41)	65
slime	400	2.68	(1.66)	5.36	(2.63)	4.17	(1.82)	1
slow	982	3.93	(1.60)	3.39	(2.22)	4.35	(1.61)	60
slum	401	2.39	(1.25)	4.78	(2.52)	3.83	(2.18)	8
slush	983	4.66	(1.88)	3.73	(2.23)	4.91	(1.48)	.
smallpox	402	2.52	(2.08)	5.58	(2.13)	4.29	(2.17)	2
smooth	984	6.58	(1.78)	4.91	(2.57)	5.09	(2.09)	42
snake	609	3.31	(2.20)	6.82	(2.10)	3.78	(2.05)	44

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)	Arousal Mean(SD)	Dominance Mean (SD)	Word Frequency
snob	403	3.36 (1.81)	5.65 (2.36)	5.11 (2.25)	1
snow	575	7.08 (1.83)	5.75 (2.47)	5.80 (1.97)	59
snuggle	404	7.92 (1.24)	4.16 (2.80)	5.66 (2.47)	4
social	985	6.88 (1.82)	4.98 (2.59)	5.91 (2.07)	380
soft	986	7.12 (1.34)	4.63 (2.61)	6.00 (1.80)	61
solemn	405	4.32 (1.51)	3.56 (1.95)	4.61 (1.87)	12
song	987	7.10 (1.97)	6.07 (2.42)	5.85 (2.12)	70
soothe	988	7.30 (1.85)	4.40 (3.08)	5.36 (2.24)	2
sour	989	3.93 (1.98)	5.10 (1.95)	4.64 (1.50)	3
space	574	6.78 (1.66)	5.14 (2.54)	5.20 (2.44)	184
spanking	990	3.55 (2.54)	5.41 (2.73)	3.91 (2.51)	.
sphere	991	5.33 (0.87)	3.88 (1.99)	5.00 (0.92)	22
spider	610	3.33 (1.72)	5.71 (2.21)	4.75 (2.11)	2
spirit	406	7.00 (1.32)	5.56 (2.62)	5.82 (2.42)	182
spouse	407	7.58 (1.48)	5.21 (2.75)	5.53 (1.97)	3
spray	992	5.45 (1.63)	4.14 (2.28)	5.12 (1.43)	16
spring	993	7.76 (1.51)	5.67 (2.51)	6.26 (1.98)	127
square	408	4.74 (1.02)	3.18 (1.76)	4.51 (1.45)	143
stagnant	994	4.15 (1.57)	3.93 (1.94)	4.71 (1.36)	5
star	409	7.27 (1.66)	5.83 (2.44)	4.68 (2.15)	25
startled	410	4.50 (1.67)	6.93 (2.24)	4.48 (1.57)	21
starving	611	2.39 (1.82)	5.61 (2.53)	3.63 (2.10)	6
statue	995	5.17 (0.70)	3.46 (1.72)	4.95 (1.40)	17
stench	996	2.19 (1.37)	4.36 (2.46)	4.29 (1.91)	1
stiff	997	4.68 (1.97)	4.02 (2.41)	4.93 (2.04)	21
stink	411	3.00 (1.79)	4.26 (2.10)	4.16 (1.98)	3
stomach	998	4.82 (2.06)	3.93 (2.49)	4.68 (1.85)	37
stool	999	4.56 (1.72)	4.00 (2.14)	4.98 (1.85)	8
storm	1000	4.95 (2.22)	5.71 (2.34)	4.54 (2.04)	26
stove	1001	4.98 (1.69)	4.51 (2.14)	5.36 (1.87)	15
street	412	5.22 (0.72)	3.39 (1.87)	4.81 (1.21)	244
stress	413	2.09 (1.41)	7.45 (2.38)	3.93 (2.75)	107
strong	414	7.11 (1.48)	5.92 (2.28)	6.92 (2.43)	202
stupid	415	2.31 (1.37)	4.72 (2.71)	2.98 (2.18)	24
subdued	416	4.67 (1.31)	2.90 (1.81)	4.08 (1.56)	8
success	417	8.29 (0.93)	6.11 (2.65)	6.89 (2.40)	93
suffocate	418	1.56 (0.96)	6.03 (3.19)	3.44 (2.81)	1
sugar	1002	6.74 (1.73)	5.64 (2.18)	5.50 (1.50)	34
suicide	419	1.25 (0.69)	5.73 (3.14)	3.58 (3.02)	17
sun	532	7.55 (1.85)	5.04 (2.66)	6.16 (2.09)	112
sunlight	1003	7.76 (1.43)	6.10 (2.30)	5.63 (2.15)	17
sunrise	420	7.86 (1.35)	5.06 (3.05)	5.29 (2.41)	10
sunset	421	7.68 (1.72)	4.20 (2.99)	5.66 (2.08)	14
surgery	612	2.86 (2.19)	6.35 (2.32)	2.75 (1.86)	6
surprised	422	7.47 (1.56)	7.47 (2.09)	6.11 (2.19)	58
suspicious	423	3.76 (1.42)	6.25 (1.59)	4.47 (1.99)	13
swamp	1004	5.14 (2.24)	4.86 (2.36)	5.29 (1.63)	5
sweetheart	424	8.42 (0.83)	5.50 (2.73)	6.03 (2.24)	9
swift	1005	6.46 (1.76)	5.39 (2.53)	6.29 (1.85)	32
swimmer	576	6.54 (1.64)	4.82 (2.49)	5.96 (1.91)	.
syphilis	425	1.68 (1.23)	5.69 (3.25)	3.33 (2.67)	.
table	426	5.22 (0.72)	2.92 (2.16)	4.47 (1.66)	198
talent	427	7.56 (1.25)	6.27 (1.80)	6.49 (1.75)	40
tamper	1006	4.10 (1.88)	4.95 (2.01)	4.58 (2.10)	1
tank	613	5.16 (1.87)	4.88 (1.86)	4.78 (1.93)	12
taste	1007	6.66 (1.57)	5.22 (2.38)	5.50 (1.65)	59
taxi	1008	5.00 (1.96)	3.41 (2.14)	4.64 (1.83)	16
teacher	1009	5.68 (2.12)	4.05 (2.61)	5.11 (2.20)	80
tease	1010	4.84 (2.51)	5.87 (2.56)	4.67 (2.37)	6
tender	1011	6.93 (1.28)	4.88 (2.30)	5.33 (1.75)	11
tennis	540	6.02 (1.97)	4.61 (2.60)	5.61 (2.12)	15
tense	428	3.56 (1.36)	6.53 (2.10)	5.22 (2.02)	15
termite	429	3.58 (2.08)	5.39 (2.43)	3.87 (1.87)	.
terrible	430	1.93 (1.44)	6.27 (2.44)	3.58 (2.34)	45
terrific	431	8.16 (1.12)	6.23 (2.73)	6.60 (2.15)	5
terrified	432	1.72 (1.14)	7.86 (2.27)	3.08 (2.75)	7
terrorist	614	1.69 (1.42)	7.27 (2.38)	2.65 (2.30)	.
thankful	433	6.89 (2.29)	4.34 (2.31)	5.32 (2.00)	6
theory	434	5.30 (1.49)	4.62 (1.94)	4.88 (1.81)	129
thermometer	1012	4.73 (1.05)	3.79 (2.02)	4.39 (1.51)	.
thief	435	2.13 (1.69)	6.89 (2.13)	3.79 (2.55)	8
thorn	436	3.64 (1.76)	5.14 (2.14)	4.45 (1.50)	3
thought	1013	6.39 (1.58)	4.83 (2.46)	6.02 (1.70)	515
thoughtful	437	7.65 (1.03)	5.72 (2.30)	5.61 (2.11)	11
thrill	438	8.05 (1.48)	8.02 (1.65)	6.54 (2.30)	5
tidy	1014	6.30 (1.56)	3.98 (2.22)	5.49 (1.93)	1
time	439	5.31 (2.02)	4.64 (2.75)	4.63 (2.24)	1599
timid	440	3.86 (1.55)	4.11 (2.09)	3.09 (1.91)	5

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
tobacco	441	3.28	(2.16)	4.83	(2.90)	4.08	(2.27)	19
tomb	442	2.94	(1.88)	4.73	(2.72)	3.72	(2.05)	11
tool	1015	5.19	(1.27)	4.33	(1.78)	5.67	(1.62)	40
toothache	443	1.98	(1.15)	5.55	(2.51)	3.90	(1.85)	.
tornado	444	2.55	(1.78)	6.83	(2.49)	4.30	(2.42)	1
torture	445	1.56	(0.79)	6.10	(2.77)	3.33	(2.37)	3
tower	1016	5.46	(1.75)	3.95	(2.28)	5.78	(2.14)	13
toxic	446	2.10	(1.48)	6.40	(2.41)	4.42	(2.51)	3
toy	1017	7.00	(2.01)	5.11	(2.84)	6.09	(1.84)	4
tragedy	447	1.78	(1.31)	6.24	(2.64)	3.50	(2.34)	49
traitor	448	2.22	(1.69)	5.78	(2.47)	4.61	(2.71)	2
trash	615	2.67	(1.45)	4.16	(2.16)	5.24	(1.85)	2
trauma	616	2.10	(1.49)	6.33	(2.45)	2.84	(1.87)	1
travel	1018	7.10	(2.00)	6.21	(2.51)	6.31	(2.08)	61
treasure	449	8.27	(0.90)	6.75	(2.30)	6.36	(2.42)	4
treat	1019	7.36	(1.38)	5.62	(2.25)	5.78	(1.82)	26
tree	450	6.32	(1.56)	3.42	(2.21)	5.08	(2.29)	59
triumph	451	7.80	(1.83)	5.78	(2.60)	6.98	(2.20)	22
triumphant	452	8.82	(0.73)	6.78	(2.58)	6.95	(2.55)	5
trophy	453	7.78	(1.22)	5.39	(2.44)	6.44	(2.32)	8
trouble	454	3.03	(2.09)	6.85	(2.03)	4.85	(2.39)	134
troubled	455	2.17	(1.21)	5.94	(2.36)	3.91	(2.33)	31
truck	577	5.47	(1.88)	4.84	(2.17)	5.33	(1.83)	57
trumpet	456	5.75	(1.38)	4.97	(2.13)	4.57	(1.72)	7
trunk	1020	5.09	(1.57)	4.18	(2.19)	5.14	(1.90)	8
trust	457	6.68	(2.71)	5.30	(2.66)	6.61	(2.04)	52
truth	458	7.80	(1.29)	5.00	(2.77)	6.47	(2.11)	126
tumor	459	2.36	(2.04)	6.51	(2.85)	3.58	(2.42)	17
tune	1021	6.93	(1.47)	4.71	(2.09)	5.74	(1.82)	10
twilight	1022	7.23	(1.80)	4.70	(2.41)	5.59	(1.82)	4
ugly	460	2.43	(1.27)	5.38	(2.23)	4.26	(2.33)	21
ulcer	461	1.78	(1.17)	6.12	(2.68)	4.17	(2.22)	5
umbrella	578	5.16	(1.57)	3.68	(1.99)	5.42	(1.91)	8
unfaithful	462	2.05	(1.55)	6.20	(2.70)	3.02	(2.54)	1
unhappy	463	1.57	(0.96)	4.18	(2.50)	3.34	(2.35)	26
unit	1023	5.59	(1.87)	3.75	(2.49)	5.11	(1.74)	103
untroubled	464	7.62	(1.41)	3.89	(2.54)	5.53	(2.54)	.
upset	465	2.00	(1.18)	5.86	(2.40)	4.08	(2.31)	14
urine	617	3.25	(1.71)	4.20	(2.18)	5.24	(1.86)	1
useful	466	7.14	(1.60)	4.26	(2.47)	5.93	(2.10)	58
useless	467	2.13	(1.42)	4.87	(2.58)	3.92	(2.62)	17
utensil	1024	5.14	(1.39)	3.57	(1.98)	5.40	(1.47)	.
vacation	468	8.16	(1.36)	5.64	(2.99)	6.80	(2.08)	47
vagina	1025	6.14	(1.77)	5.55	(2.55)	5.88	(1.74)	10
valentine	469	8.11	(1.35)	6.06	(2.91)	5.81	(2.45)	2
vampire	470	4.26	(1.86)	6.37	(2.35)	5.05	(2.27)	1
vandal	471	2.71	(1.91)	6.40	(1.88)	3.91	(2.49)	1
vanity	472	4.30	(1.91)	4.98	(2.31)	4.80	(2.03)	7
vehicle	473	6.27	(2.34)	4.63	(2.81)	5.77	(2.61)	35
venom	474	2.68	(1.81)	6.08	(2.44)	3.94	(2.23)	2
vest	1026	5.25	(1.33)	3.95	(2.09)	5.09	(1.24)	4
victim	618	2.18	(1.48)	6.06	(2.32)	2.69	(2.04)	27
victory	475	8.32	(1.16)	6.63	(2.84)	7.26	(2.14)	61
vigorous	476	6.79	(1.54)	5.90	(2.66)	5.41	(2.22)	29
village	477	5.92	(1.34)	4.08	(1.87)	4.94	(1.74)	72
violent	478	2.29	(1.78)	6.89	(2.47)	5.16	(2.86)	33
violin	579	5.43	(1.98)	3.49	(2.26)	5.18	(2.01)	11
virgin	1027	6.45	(1.76)	5.51	(2.06)	6.24	(2.48)	35
virtue	479	6.22	(2.06)	4.52	(2.52)	6.13	(2.09)	30
vision	480	6.62	(1.84)	4.66	(2.43)	6.02	(1.96)	56
volcano	619	4.84	(2.14)	6.33	(2.21)	3.25	(1.97)	2
vomit	481	2.06	(1.57)	5.75	(2.84)	3.58	(2.45)	3
voyage	1028	6.25	(1.91)	5.55	(2.23)	5.18	(1.98)	17
wagon	1029	5.37	(0.97)	3.98	(2.04)	5.05	(1.20)	55
war	482	2.08	(1.91)	7.49	(2.16)	4.50	(3.00)	464
warmth	483	7.41	(1.81)	3.73	(2.40)	5.61	(1.67)	28
wasp	484	3.37	(1.63)	5.50	(2.17)	3.76	(1.82)	2
waste	485	2.93	(1.76)	4.14	(2.30)	4.72	(1.94)	35
watch	580	5.78	(1.51)	4.10	(2.12)	5.37	(1.75)	81
water	486	6.61	(1.78)	4.97	(2.49)	5.08	(1.99)	442
waterfall	487	7.88	(1.03)	5.37	(2.84)	5.20	(2.18)	2
wealthy	488	7.70	(1.34)	5.80	(2.73)	6.77	(2.57)	12
weapon	489	3.97	(1.92)	6.03	(1.89)	5.19	(2.61)	42
weary	490	3.79	(2.12)	3.81	(2.29)	4.00	(1.91)	17
wedding	491	7.82	(1.56)	5.97	(2.85)	6.68	(2.08)	32
whistle	1030	5.81	(1.21)	4.69	(1.99)	5.27	(1.87)	4
white	542	6.47	(1.59)	4.37	(2.14)	5.98	(1.73)	365
whore	492	2.30	(2.11)	5.85	(2.93)	4.61	(2.73)	2

Affective Norms for English Words. **All Subjects** *Table 1*
Bradley, M.M., & Lang, P.J. (1999)

Description	Word No.	Valence Mean(SD)		Arousal Mean(SD)		Dominance Mean (SD)		Word Frequency
wicked	493	2.96	(2.37)	6.09	(2.44)	4.36	(2.65)	9
wife	1031	6.33	(1.97)	4.93	(2.22)	5.57	(1.68)	228
win	494	8.38	(0.92)	7.72	(2.16)	7.39	(2.36)	55
windmill	1032	5.60	(1.65)	3.74	(2.13)	5.24	(1.04)	1
window	495	5.91	(1.38)	3.97	(2.01)	4.91	(1.60)	119
wine	496	5.95	(2.19)	4.78	(2.34)	5.31	(2.15)	72
wink	1033	6.93	(1.83)	5.44	(2.68)	5.70	(1.77)	7
wise	497	7.52	(1.23)	3.91	(2.64)	6.70	(2.39)	36
wish	1034	7.09	(2.00)	5.16	(2.62)	5.28	(2.09)	110
Wit	1035	7.32	(1.90)	5.42	(2.44)	6.38	(2.01)	20
Woman	498	6.64	(1.76)	5.32	(2.59)	6.33	(1.52)	224
Wonder	499	6.03	(1.58)	5.00	(2.23)	5.32	(2.17)	67
world	500	6.50	(2.03)	5.32	(2.39)	5.26	(2.47)	787
wounds	620	2.51	(1.58)	5.82	(2.01)	3.92	(1.57)	8
writer	1036	5.52	(1.90)	4.33	(2.45)	4.73	(1.84)	73
yacht	1037	6.95	(1.79)	5.61	(2.72)	6.10	(2.13)	4
yellow	545	5.61	(1.94)	4.43	(2.05)	5.47	(1.58)	55
young	1038	6.89	(2.12)	5.64	(2.51)	5.30	(2.49)	385
youth	1039	6.75	(2.29)	5.67	(2.52)	5.11	(2.55)	82
zest	1040	6.79	(2.04)	5.59	(2.66)	6.00	(1.99)	5

3. DETECTING THE EMOTION OF A SENTENCE

First, word segmentation, POS annotation and NE recog-nition are performed for lyrics, with the help of the NLP tool. After stop words removed, the remaining words of a sentence are examined to see if they appear in ANCW, and each of the words that do appear in ANCW constitutes an EU. If there is an adverb that modifies or negates an emotion word, it is included in the corresponding EU as a modifier. We recognize the modifiers of EUs by using the NLP tool. The emotion of an EU is determined as follows:

$$vu = vWord(u) \cdot mModifier(u),v \quad (1)$$

$$au = aWord(u) \cdot mModifier(u),a \quad (2)$$

The lexicon of synonyms is manually built and includes 77,343 terms

Where v_u and a_u denote the valence and arousal value of EU u respectively, $v_{Word(u)}$ and $a_{Word(u)}$ denote the valence and arousal value of the EU's emotion word re-spectively, $m_{Modifier(u),v}$ and $m_{Modifier(u),a}$ denote mod-ifying factors to represent the effect of the EU's modifier on the EU's valence and arousal respectively.

and $a_{Word(u)}$, the valence and arousal value of the emo-tion word are obtained through looking up in ANCW. Sen-tences that have not any emotion unit are discarded.

We have collected 276 individual modifier words, which cover all the occurrences in the lyric corpus we use, and a table of modifiers has been set up. According to the polarities and degrees to which modifiers influence the emotions of EUs, we assign each modifier a modifying fac-tor on valence and a modifying factor on arousal. The val-ues of the modifying factors are in the range of [−1.5, 1.5].

For a negative modifier adverb, $m_{Modifier(u),v}$ is set to a value in [−1.5, 0] and for a positive modifier adverb, it is set to a value in [0, 1.5].

14 INTEGRATING THE EMOTIONS OF ALL SENTENCES

4.1 Challenges

3. Reduce the effect of errors in sentence emotions on the result of the emotions of lyrics.

4. Recognize all the emotions of a lyric on the condi-tion that the lyric has more than one emotion.

5. Select one emotion as the main emotion, if needed, or give a probability to each of the emotions.

4.2 Methodology

In recent years, spectral clustering based on graph parti-tion theories decomposes a document corpus into a num-ber of disjoint clusters which are optimal in terms of some predefined criteria functions. If the sentences of a lyric are considered as documents and the lyric is regarded as the document set, the document clustering technology can conquer the above three challenges. We define an emotion vector space model, where each sentence of a lyric is con-sidered as a node with two dimensions that represent the valence and arousal of an emotion respectively. We choose Wu's fuzzy clustering method [12] because it can cluster the sentences without the need to specify the number of clusters, which meets our demands. Wu's fuzzy cluster-ing method includes three steps: building a fuzzy similar-ity matrix, generating a maximal tree using Prim algorithm and cutting tree's edges whose weight is lower than a given threshold.

A song usually repeat some sentences. Sometimes the repeated sentences are placed in one line, with each sen-tence having its own time tag. In other cases, each repeated sentence occupies one line and the line has one time tag. If the repeated sentences are placed in more than one lines, these sentences are bound to form a cluster in the later clustering processing. If the emotions of those repeated sentences were not recognized correctly, subsequent pro-cessing will be ruined definitely. Hence, before sentences are clustered, lyrics should be compressed so as to place the iterative sentences in one line, with each sentence hav-ing its own time tag.

Having examined hundreds of lyrics, we find that sen-tences in a lyric always fall into several groups. The sen-tences of a group have similar emotions which can be uni-fied to a prominent emotion of the lyric. Therefore, the isolated sentences are mostly noises and will be removed.

There are a dozen of means to measure the similarity be-tween two nodes in vector space. After experiment those means, we select the following means to measure the sim-ilarity of the sentences' emotions *i, j*.

$$Sim_{ij} = 1 - \sigma(|v_i - v_j| + |a_i - a_j|) \qquad (7)$$

where v_i,v_j,a_i, and a_j denote the valence and arousal of sentences *i* and *j* respectively, and σ is set to 0.3.

The center of a survived cluster is calculated as the weighted mean of emotions of all members of the cluster. The weighted mean is defined as follows:

$$v_c = \frac{\sum_{s \in S_c} v_s \cdot w_s}{|S_c|} \qquad (8)$$

$$a_c = \frac{\sum_{s \in S_c} a_s \cdot w_s}{|S_c|} \qquad (9)$$

where S_c denotes the set of sentences in cluster c, v_c and a_c denote the valence and arousal respectively of cluster c, and v_s, a_s and w_s denote the valence, arousal and weight respectively of sentence $s (s \in S_c)$.

The weight of cluster c is calculated as follows:

$$w_c = \sum_{s \in S_c} \frac{(\alpha \cdot w_s + \beta \cdot Loop(s))}{-\gamma \cdot r_s + 1} \qquad (10)$$

where $Loop(s)$ denotes the number of times sentence $s (s \in S_c)$ repeats, α, β and γ are set to 2, 1, 1, respec-tively. These constant parameters are adjusted through ex-perimentation and the set of values resulting in the highest F-measure was chosen.

Lyrics we got have time tags and we use these tags to compute the singing speed of sentences in lyrics, which is defined in milliseconds per word. Although, singing speed is not the only determinant of the emotions of lyrics, there is correlation between the singing speed of a song and its emotions, as shown in Figured 5. Hence, we use singing speeds of sentences to re-weight each clustering center. Having analyzed the singing speeds and emotions of the songs in the corpus, we think that Gaussian Model is suit-able for expressing the degrees to which different singing speeds influence emotions.

5. Experimental Result

In this section we present the main decisions taken throughout the development of this project. In what concerns to MARSYAS, this covers aspects related to feature extraction, data normalization, classification, playlist generation and evaluation. It also includes the back office part of the application, developed in Qt.

AUDIO BASED APPROACH:

5.1. Feature extraction

To achieve the main purpose of the developed application, the first step was clearly to extract the features of the all the songs on the dataset (the 194 songs samples annotated by Yang). All the features sets provided by MARSYAS were used (13 features sets), without selection, which comprises 454 features in total for each song to be extracted. This number includes statistical features derived from the core ones (means, standard deviations, etc), among others:

- Tempo
- Stereo Panning Spectrum Features
- Mel Frequency Cepstral Coefficients
- Chroma
- Spectral Flatness Measure
- Spectral Crest Factor
- Spectral Centroid
- Spectral Rolloff
- Spectral Flux
- Line Spectral Pair
- Linear Prediction Cepstral Coefficients
- Zero Crossings
- Beat

Since the objective was to classify each aforementioned sample with a single AV (arousal/valence) pair (which would represent the value for the whole song segment), and consequently a unique value for each feature depicting the entire song, a single feature of values was used (MARSYAS provides the possibility of using other kind of networks, which can extract many values for a given song feature – for instance, for mood tracking purposes, among others).

5.2. Data normalization

Between feature extraction and classification, classification data (*i.e.* feature extraction values) was normalized, to ensure that all values ranged between the same boundaries, preventing the classification results to become corrupted. where is the normalized value, l and u are respectively the upper and lower limits between which the features values will be scaled. In this case, the chosen feature normalization interval was [0, 1] (u=0 and l=1).

5.3. Classification

To properly classify the songs and estimate distances between them, both arousal and valence values needed to be known simultaneously. With this in mind, the initial method developed by my colleague Renato Panda was improved – so that the mentioned pair values could be predicted at the same time. Once again (as in Renato's thesis), the SVM classifier was used for this, through the libSVM library, on the Yang dataset (as referred earlier in this report, 194 samples of songs, annotated with arousal and valence values).

It was chosen to use K-fold cross-validation, with K=4, which means that 3 folds were used for the training phase (75% of the dataset), while the last fold was used for testing (25% of the dataset). This means that each fold would have 48 or 49 songs (considering the 194 that comprised the whole dataset). As this cross-validation method implies, all the 4 folds were rotated, to ensure that all of them were used for both training and testing purposes.

The aforementioned process was repeated 50 times, which means that 200 folds (4 x 50) were generated. It is also important to underline that all the folds were randomly generated.

5.4. Database

A SQL database is used by server, which stores all the information about classification and the songs. The server does all the operations required on the database (e.g. querying, update and addition of data). The use of Qt SQL libraries provides support for different database management systems, also supporting different engines, from a simple SQLite file to MySQL, PostgreSQL, Oracle or Acess DB files or any other Open Data Base Connectivity (ODBC) protocol. Currently, the prototype supports SQLite, with preliminary support for MySQL.

The database was planned and designed in a general and expandable way. It supports the current needs but also allows different mood models, including different types (both categorical and dimensional), user accounts, lists of artists, albums, genres, features and classification profiles, saving different classification and tracking values for the same song, based on different combinations of features and classifiers, for instance.

5.5. Song Details

After selecting one song in the database map, a request for its details is made to the server. The information is then received and displayed on the “Song details” dialog (Figure 24), which displays all the ID3 tag information data available for the specified song, as long as the correspondent soundwave graph and mood tracking data (based on Renato’s work). The song can be played and a vertical black line marks the progress of it, with different colors representing different mood quadrants identified in the song.

5.6.Song annotations

As pointed out in previous sections of this report, the song dataset used was kindly provided by Yi-Hsuan Yang, one of the authors of (Yang, Lin, Su, & Chen, 2006), along with the arousal and valence annotations values for each one of the dataset songs. This dataset was the one use for training and testing purposes, and consisted of 194 songs from several genres and provenances, spanning different arousal and valence values, from all four Thayer’s model quadrants.

However, as my colleague Renato Panda pointed out in his thesis, the annotations values revealed that they aren’t 100% accurate and not as diverse as one would expect, which of course can negatively influence the final results. This will be addressed below.

5.7. Proximity to Thayer’s model origin

After mapping the dataset annotations to the Thayer’s model, it is obvious that the majority of the songs are close to the origin of both axes, which suggests that they don’t denote very marked moods (if that was the case they would be much more distant from the axes origin, at least the majority). This can be seen both in Figure 29 and Table 5, below:

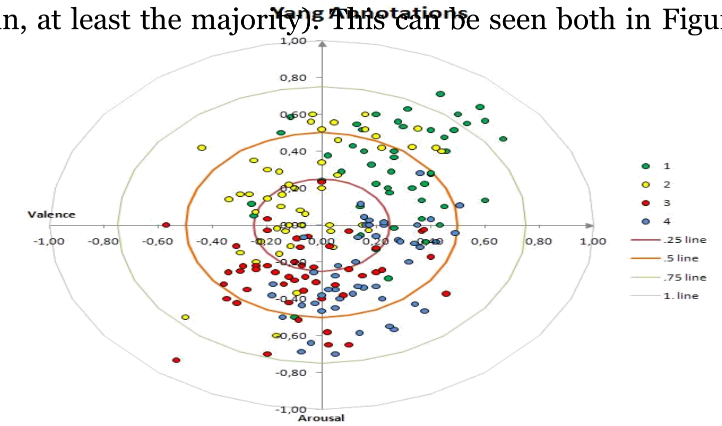


Figure 29: Yang annotations mapped into Thayer's model[19]

Distance from the origin	Number of songs	Percentage in the dataset	Sum
[0, 0.25]	47	24.23 %	47
]0.25, 0.5]	93	47.94 %	140
]0.5, 0.75]	47	24.23 %	187
]0.75, 1]	7	3.61 %	194

Table 5: Yang annotations distances to the model's origin[20]

As it can be seen in Table 5, nearly 25% of the songs annotations are within a distance of 0.25 of the center (red circumference in Figure 29), while nearly 75% of the dataset is at most at a distance of 0.5 from the graph's origin (orange circumference in Figure 29). This means that the majority of the songs are placed near the model's origin, which obviously leads to somewhat ambiguous moods.

5.8. Unbalanced song distribution

The Yang dataset aimed to achieve a balanced distribution of songs in the four quadrants that compose the Thayer's Model. Since the dataset has 194 songs, this would mean about 48 or 49 songs for each quadrant. After analyzing the annotations values, it is quite evident that this differ much from the initial quadrants (expressed in each song filename), which results in an unbalanced dataset.

Again, looking to Figure 29 (where each one of the four colors represents one of the four quadrants), it was expected to see the points with the same color (the initial annotation) together in the same quadrant. However, as it can be seen in Figure 29 and analyzed in Table 6, it is quite obvious that the songs are scattered all over the model's quadrants, which makes the dataset completely unbalanced. This is another issue that can also lead to distorted results

Quadrant	Yang	Real	Real (%)	Matching	Matching (%)
1	48	54	27.84 %	36	75.00 %
2	48	22	11.34 %	17	35.42 %
3	49	51	26.29 %	16	32.65 %
4	49	49	25.26 %	9	18.37 %
Other	0	18	9.28 %	-	-
Total	194	194	100.0 %	78	40.21 %

Table 6: Songs per quadrant (Yang's vs. real annotations)[20]

In Table 6, the "Yang" column shows the number of songs in each quadrant, according to Yang's initial annotations, while the "Real" column shows how many songs were indeed in each quadrant, after the analysis of the annotated values (in this particular case, the "Other" row shows 18 songs where one of the values for arousal or valence was equal to zero, therefore not belonging to any particular quadrant). The "Matching" column shows how many songs were indeed correct (matching in both Yang's initial annotations and the annotated values).

In a quick glance, it can be seen that only nearly 40 % of the songs (78 songs) match both annotations, and that the second quadrant actually only has 22 songs in it.

Song Title	Artist	Emotion
1. LDN	Lily Allen	Exuberance
2. Dancing Queen	ABBA	Exuberance
3. Shiny Happy People	R.E.M.	Exuberance
4. Love Shack	B52S	Exuberance
5. She Loves You	Beatles	Exuberance
6. Crazy in Love	Beyoncé	Exuberance
7. Baby One More Time	Britney Spears	Exuberance
8. I've had the Time of my Life	Bill Medley	Exuberance
9. Spice Up Your Life	Spice Girl	Exuberance
10. Spinning Around	Kylie Minogue	Exuberance
11. Fight Song	Rachel Pattern	Anxious
12. Mad Hatter	Melaine Martinez	Anxious
13. Home	Meg Hutchinson	Anxious
14. Unwell	Matchbox Twenty	Anxious
15. Warrior	Demi Lovato	Anxious
16. The Middle	Jimmy Eat World	Anxious
17. Breathe Me	Sia	Anxious
18. About Today	The National	Anxious
19. Shake It Out	Florence + Machine	Anxious
20. Drown	Bring Me the Horizon	Anxious
21. Migraine	21 Pilots	Anxious
22. Amazing Day	Coldplay	Contentment
23. Be my Forever	Christina Perri	Contentment
24. Best of Joy	Michel Jackson	Contentment
25. Brighter than the Sun	Colbie	Contentment
26. Do Everything	Steven Curtis Chapman	Contentment
27. Everybody Have Fun Tonight	Wong Chong	Contentment
28. For Once in My Life	Stevie Wonder	Contentment
29. Good Time	Owl City	Contentment
30. If You Wanna be Happy	Jimmy Soul	Contentment
31. It Was a Good Day	Ice Cube	Contentment
32. Angels	Robbie Willams	Depression
33. Wish You Were Here	Pink Floyd	Depression
34. Breathe	Ana Nalick	Depression
35. Under the Bridge	Red Hot Chili Peppers	Depression
36. Creep	Radiohead	Depression
37. Wicked Gone	Chris Isaak	Depression
38. Brick	Ben Folds Five	Depression
39.SomeThing in the Way	Nirvana	Depression
40. Time After	Cyndi Lauper	Depression
41. Unchanged Melody	Righteous Brothers	Depression

5.9. Global results

The results obtained by the tests specified in section 3.1.5. Playlist evaluation are presented in Table 7. It is important to know that these results are the arithmetic means of each one of the metrics specified, based on all the measures made (9700 songs = 50 repetitions × 4 folds × all songs on each test fold – 2 folds have always 49 songs, while the other 2 have always 48):

Metric	Results (arithmetic mean)
Playlist first song match (size 20)	4.11 %
Percentage of playlist songs match (size 5)	19.03 %
Percentage of playlist songs match (size 10)	35.35 %
Percentage of playlist songs match (size 20)	58.73 %

Table 7: Experimental results

From this table, it is easy to understand why the exposed metrics tests results grow in percentage.

The first metric has undoubtedly a really low result, but that can be somewhat understand if we have in mind that it measures the average percentage of times the closest song in a top distance list of 48 or 49 songs is the same in the annotation and prediction ones.

The other three metrics are similar among themselves, the only variable being the playlist size. Since these three metrics measure the average percentage of common songs (therefore ignoring the order of appearance of them) between the top annotation and top prediction lists (with playlists sizes of 5, 10 and 20 songs), it is easy to understand that, the bigger the playlist size is, the most probable is to find common songs in the two top lists. To a playlist of size 20, in these conditions, the results are reasonable, with an average matching of almost 60%.

However, these results also predict that if the dataset (and consequently the fold) size is increased, these four metrics values will probably drop. The limitations that helped led to his results were already pointed out and some suggestions are made abaixo (section 5 Future work).

5.10. Classification results

However, it is noteworthy that the fact that feature selection was not performed was another factor that contributed to the somewhat low results achieved. In his thesis, using all the features available in MARSYAS, my colleague Renato Panda concluded that, as previous studies pointed out, valence values are easier to predict, in comparison with the valence ones. His tests used R2 and RMSE statistics (see section 3.1.3. Classification for their definition) to verify this and indeed they revealed that the R2 value for arousal reached 57.9% but only 3.24% for valence (Table 8). The arousal values are similar to the ones observed in (Yang, Lin, Su, & Chen, 2006), but in what concerns with the valence values, they reached 28.1% in the same paper, much more than the ones obtained here. This discrepancy is probably due to the fact that, in the aforementioned paper, several feature extraction frameworks were used, which provided the authors with a wider range of features, some of them not present in MARSYAS. This is underlined by the fact that three out of the four most important features pointed out by Yang in his paper are no present in MARSYAS.

We also conducted a pilot study with MIR Toolbox, which led to a R2 value of 25% for valence, which confirms the absence of some meaningful features in MARSYAS. Another cause to this low R2 results for valence probably had to do with the use of the entire MARSYAS feature set. If feature selection had been made, some features (with uninteresting results for mood classification) would be discarded, thus increasing both the results (especially for valence) and the model accuracy.

Arousal				Valence			
SSE	SST	RMSE	R^2	SSE	SST	RMSE	R^2
0.996816	2.34311	0.220108	0.57985	1.15312	1.1943	0.241297	0.0324472

Table 8: Global classification results (using all features)

Also, looking at the placement of the predictions on Thayer's model it is easy to conclude that all songs are gathered within the 0.50 distance limit (orange circumference, see Figure 30) and that the valence values practically do not change (a very small variation almost places them all over the arousal axis).

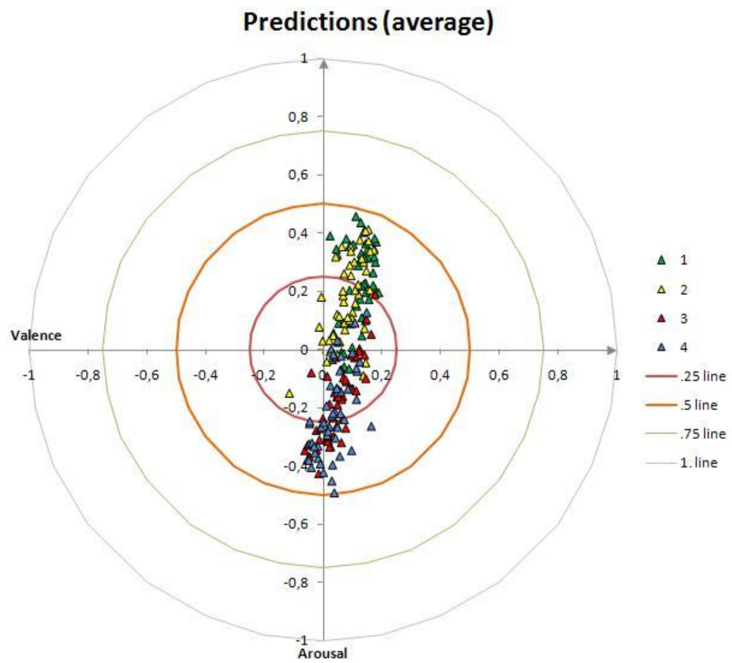


Figure 30: Global predictions in Thayer's model (using all features)[19]

As previously pointed out, improvements to these results can be achieved by adding meaningful features, such as tonality, multiplicity, spectral dissonance and chord, as mentioned in (Yang, Lin, Su, & Chen, 2006) (other suggestions are exposed in section 5.1. Future work).

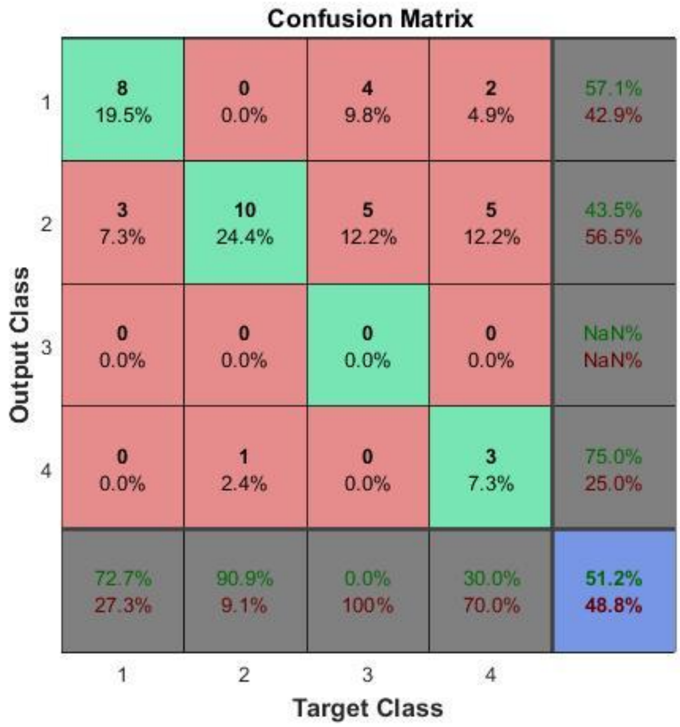
LYRIC BASED APPROACH:

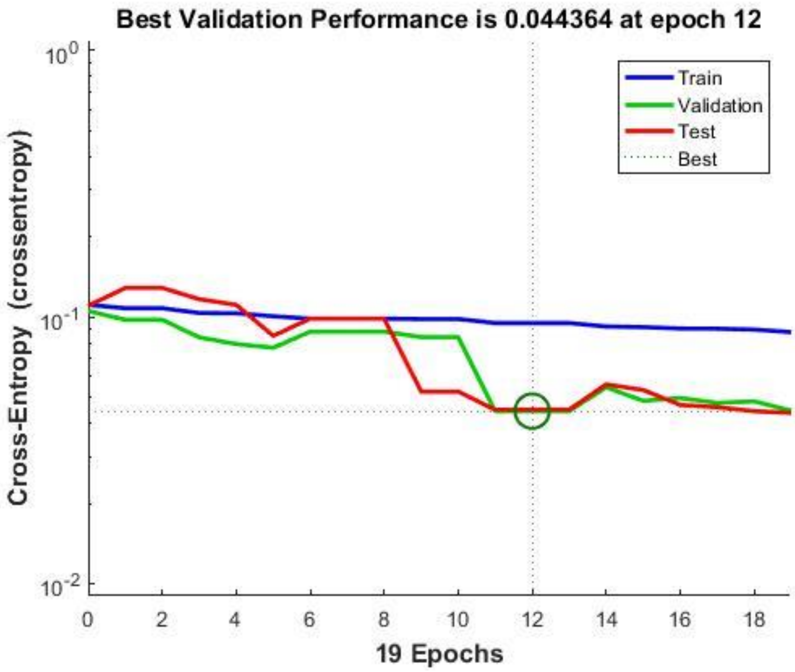
Our ultimate goal is to compute the valence and arousal value of lyrics, not to do classification. We do classification for broad classes for the purpose of evaluating our emotion detecting method and comparing the performance of our method with that of other classification methods proposed in the literatures, many of which were for the same broad classes.

5.11 Data Sets

To evaluate the performance of our approach, we collected 42 songs from the classified catalogue accord-ing to emotion in www.koook.com. These songs are up-loaded by netizens and their genres include pop, rock & roll and rap. These songs were labeled by 7 people whose ages are from 23 to 48. Two of them are professors and five are postgraduate students, all native Chinese. Each judge was asked to give only one label to a song. The songs that are labeled by at least 6 judges to the same class are re-mained. We use these songs' lyrics as the corpus. The distribution of the corpus in four classes is shown in Table

☐ Although the number of songs in +V-A class is small, it is not surprising. This phenomenon conforms to the distri-bution in reality.





5.12 Results

To demonstrate how our approach improves the emotion classification of lyrics in comparison to existing methods, we implemented a emotion classification method based on lyrics with emotion lexicon: Lyricator [10]. Lyricator uses ANEW to extend the emotion lexicon by natural language corpus with a co-occurrence method. Using the extended emotion lexicon, Lyricator computes the emotion of each sentence of a lyric and the sentence emotion is the mean of emotion values of the emotion words contained in the sen-tence. The emotion of a lyric is weighted mean of values of the emotions of sentences. The weight is defined as the loop of sentences in the lyric.

To process lyrics, we translate the lexicon used in Lyricator and implement Lyricator's method. What's more, the parameters are adjusted to gain its best perfor-mance. Under the same test corpus that has been men-tioned above, we compare Lyricator with our system. Ta-ble 4 shows the evaluation results between Lyricator and our work in the same songs corpus. The precision for a class is the number of lyrics correctly labeled the class di-vided by the total number of lyrics labeled as belonging to the class. The Recall is defined as the number of true positive divided by the total number of lyrics that actually belong to the positive class. The small number of lyrics in +V-A leads to the low precision for this class. Because we have used the wealth of NLP factors and fuzzy cluster-ing method, our method's performance is better than the previous work.

Merging Factors:

- Lyrics-focused approaches are more accurate for sad songs (-ve Y axis of Thayer's Model)

- Audio-focused approaches are more accurate for happy songs (+ve Y axis)

Main Song Emotion	Audio Based Emotion	Lyrics Based Emotion
Sad	Sad	Sad
Happy	Happy	Happy
Happy	Happy	Sad
Sad	Happy	Sad

- Reasons For Result Varying:

1. Shortage of Dataset
2. Not Using NLP Tools

- Assumed Percentage of variance: 17%-25%

6. Related Works:

- ***Automatic Mood Detection and Tracking of Music Audio Signals***
- *Contributions:*
  - Music Mood Model Taxonomy description
  - Extracting music features
  - Mood Detection Algorithm
- *Limitations:*
  - Insufficient music feature
  - No relation with lyrics
  - Inefficient Mood Detection Algorithm
- ***LYRIC-BASED SONG EMOTION DETECTION WITH AFFECTIVE LEXICON AND FUZZY CLUSTERING METHOD***
- ***Contributions:***
  - Mood Model Taxonomy description
  - Extracting lyric (NLP based) features
  - Mood Detection Algorithm
- ***Limitations:***

- ❖ Insufficient ANEW list
- ❖ No relation with Music (only related with song speed)

7. Conclusions

This project revealed to be a valuable one, consisting in a valid study, with interesting and credible results. Personally, the main goals and purposes of this work were achieved, and one hint of the relevance of these results on the MIR and MER fields is the possibility of writing (or, at least, to take part on) a paper about the subject.

Of course that there were time limits and some difficulties, the main ones being the complexity of the MARSYAS core code (which increased the learning curve), and its instability, due to constant updates that sometimes generated memory leaks in some versions and so, the work done is a functional prototype of the planned mood application and not a final version.

Yet, the features that weren't developed so far are well documented in both section 5.1 and Appendix A. This way, it is easy to understand what is already done and what paths can be followed.

It is also noteworthy all the study and knowledge gained through the planning and development of this project, which is well documented in this thesis report. This consists in a solid base and introduction to the study of the MIR and MER fields. Also, despite being mainly a research project, software engineering techniques were also used, which ensured planning and eased team work and task distribution among all the members of the project.

In conclusion, this project was undoubtedly worthy and added value to the areas covered. Its results are another small but important step in the research of these recent fields (MIR and MER), and so, more data is available to all the researchers on the subject.

7.1. Future work

In a research project of this kind, usually many improvements can be done, and this one is no exception. Mainly due to time restrictions, some of the requirements planned weren't developed, although the main objectives were achieved.

In what concerns to MARSYAS and playlist generation, some of the improvements could include the use of a bigger, balanced dataset (instead of the Yang one), or, in alternative, to assure that both the training and test sets of the current dataset are balanced, including all the folds (with an equal number of songs for each quadrant). However, the latter option would mean that only 22 songs from each quadrant would be used in total (since the annotations analysis revealed that only 22 songs belonged to the second quadrant, the one with less number of songs).

Also, more metrics could be studied (which includes distance metrics, like the Euclidean and Manhattan distance between each songs feature vector, or a membership-like feature vector distance) – for instance, extract more playlist generation statistics by changing the number of repetitions and/or folds and comparing the results. Other important tool would be the use of FFS, to eventually increase the quality of the results obtained. FFS would possibly provide better results by choosing the optimal set of features (the ones that provided the best prediction results). Last, but not least, other classifiers (k-NN, GMM) could be used to compare results, and their parameters studied, compared and tuned (including the classifier used – SVM), to achieve better results.

In what concerns to the application (the client, backoffice and server), it could be improved to include all the options planned in the requirement analysis document (see Appendix A) that weren't developed. In what concerns to the database, the one currently specified and implemented is prepared for generalization and includes all the current necessary features.

Finally, at least the playlist generation experiments, metrics and results, could be documented on an article paper written on the subject.

References

Aucouturier, J.-J., & Pachet, F. (2002). Music Similarity Measures: What's The Use?

Audio Music Mood Classification Results. (s.d.). Obtido em 20 de January de 2010, de Audio Music Mood Classification Results - MIREX 2008: http://www.music-ir.org/mirex/2008/index.php/Audio_Music_Mood_Classification_Results#MIREX_2008_Audio_Mood_Classification_Run_Times

de Cheveigné, A., & Kawahara, H. (2002). YIN, a fundamental frequency estimator for speech and music. *Acoustical Society of America* .

Euclidean distance. (s.d.). Obtido em 22 de January de 2010, de Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki/Euclidean_distance

jAudio. (s.d.). Obtido em 19 de January de 2010, de jAudio:

http://jmir.sourceforge.net/jAudio.html

Laar, B. v. (2005). Emotion detection in music, a survey.

Lartillot, O. (2008). MIRtoolbox 1.1 User's Manual. Jyväskylä, Finland.

Lartillot, O., Toiviainen, P., & Eerola, T. (s.d.). *Department of Music: MIRtoolbox*. Obtido em 20 de January de 2010, de Jyväskylä University: https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox

Lu, L., Liu, D., & Zhang, H.-J. (2006). Automatic Mood Detection and Tracking of Music Audio Signals. *IEEE Transactions on Audio, Speech and Language Processing ,*

14 (1).

Marsyas. (s.d.). Obtido em 20 de January de 2010, de About:

http://marsyas.info/about/overview

McKay, C. (s.d.). jAudio: Towards a standardized extensible audio music feature extraction system.

Meyers, O. C. (2007). *A Mood-Based Music Classification and Exploration System*.

Overview. (s.d.). Obtido em 20 de January de 2010, de Marsyas:

http://marsyas.info/about/overview

Paiva, R. P. (2006). *Melody Detection in Polyphonic Audio*.

Pampalk, E. (2005). Tutorial: Music Similarity. *ISMIR*.

Pauws, S., & Eggen, B. (2002). PATS: Realization and User Evaluation of an Automatic Playlist Generator.

Ribeiro, B. (2009). Pattern Recognition Techniques slides.

Scherrer, B. (2007). Gaussian Mixture Model Classifier