Using character valence in computer generated music to produce variation aligned to a storyline

Coral Featherstone
Meraka Institute
Gauteng, South Africa
cfeatherstone@csir.co.za

Etienne Van der Poel
University of South Africa
Gauteng, South Africa
evdpoel@unisa.ac.za

ABSTRACT

This paper is to describes a method for interposing computer generated melody with tone linked to unique entities within the text of a novel.

Background: A recent study describing a piece of software called "TransProse" has already shown that sentiment in the text of a novel can be used to automatically generate simple piano music that reflects the same sentiment as the novel. This study wished to establish a method whereby, if after aligning the text with the melody, the sentiment in the words surrounding particular characters as they occurred within the novel could produce another melody line, for each character, that could reflect the individual characters' tone and distinguish the melodies ascribed to each character from each other.

Method: The sentiment in the text of the novel is extracted by looking up the words in a database that groups the words into emotional groups called "Ekman categories".

Simplistic relations between aspects of music such as pitch and tempo are chosen based on the two categories that contained the most words. These chosen attributes are then used to generate the first two melody lines.

The paragraphs within which the named entities referring to characters are found is manually determined and the top "Ekman category" of the named entities is obtained through simplistic methods of extraction. Each bar of the melody is aligned with individual paragraphs of text and an additional melody line is generated for each character.

Results: Adjusting the fitness function of the Genetic algorithm (GA) that was used was not sufficient to link the tone of the characters to the melody. Assigning each character their own short melodic phrase and varying the phrase appropriately achieved the desired outcome but requires additional work to harmonise better with the first two melody lines.

CCS Concepts

•Theory of computation → *Evolutionary algorithms;* **•Computing methodologies** → **Genetic algorithms; •Applied computing** → **Sound and music computing;**

SAICSIT '15, September 28-30, 2015, Stellenbosch, South Africa

DOI: http://dx.doi.org/10.1145/2815782.2815801

Keywords

Applications of Machine Learning, Music, Artificial Intelligence, Sentiment analysis, Opinion mining, Natural Language Processing

1. INTRODUCTION

This paper describes a system that computationally generates music whose tone is is aligned with the appearance of characters as they occur in the text of a novel.

This work is modelled on the work of Davis and Mohammad [10]. They took the electronic text of novels and counted the words associated with eight semantic categories, known as affect categories, and used the ratios of these counts to generate three melody lines. The rules of music generation, such as choice of major or minor, melody, octave choice and other musical elements were determined by the highest scoring affect.

Davis and Mohammad [10] have already shown that the affect categories of text can be used to influence the known relations between elements of music and the emotions they induce in order to align a melody with the storyline of a novel.

2. LIMITATION AND PREPARATION

The scope of this project is significant because trying to tackle everything required would necessitate domain expertise in music, computing as well as psychology. The scope is constrained by keeping the focus on the application of the computer algorithm to melody creation. Neither the complexity of introducing emotion and tone into the music nor the complexity of data mining for aspects of personality or emotive content is acknowledged at more than a superficial level. In addition no focus is given to the efficiency of the chosen algorithm.

The initial two melody lines need to be in place prior to the attempt at introducing the character specific melodies. The location of the entities in the text needs to be extracted and identified in a similar manner to Davis and Mohammad [10]. This is described in Section 4.2. The meaning of the terminology used when describing these affect categories and a shallow description of music terminology will be briefly presented in Section 4.1, providing context to what is being described. Aspects of how to represent those characteristics in the music is identified from music theory and prior research in Section 4.3.

In order to keep the music generation as simple as possible only pitch and duration was considered. Other aspects of the music such as musical form, key signatures and key were pre chosen or omitted as needed. The melodies needed to agree with each other harmonically, but no attempt was made to achieve perfect harmonic resolutions or be too prescriptive as to what musical rules could be learned.

A suitable algorithm for producing the melody lines is chosen from the field of artificial intelligence based on what has previously been shown to be successful at the task. The literature describing algorithms currently used is presented in Section 4.4.

It is only at that point that the introduction into the music of additional synchronous melody lines can be described.

3. METHODOLOGY

This process will follow the design science research methodology (DS) [25]. Unlike natural science, which tries to understand reality, DS tries to create artifacts that serve human purposes [26]. This methodology is applicable to information systems research in that it helps to elevate it from a design process into a form that can be recognised as quality research and that can be evaluated for validity and value. It does this by providing a consistent model and process to follow with specific guidelines on what the output should look like. DS follows an iterative lifecycle where it:

1. Identifies problem situations and motivates their importance, accessibility and suitability

2. Reviews existing knowledge while focusing on understanding the problem and understanding solution objectives

3. Designs a solution. (This could be a model, method or artifact)

4. Tests or demonstrates the applicability and usefulness of the result

The applicability of this paper would require the demonstration of coordination of melody lines with the novel text, and the appearances of the characters within the text. In addition, the melody lines acknowledging the characters need to be distinguishable from each other. A visual or audible indication of this should be sufficient.

4. EXISTING LITERATURE

4.1 Terminology

4.1.1 Affect categories

Emotion words are the subset of words that are considered to be emotive [28]. Valence can also be addressed on more than one level. Positive, negative or neutral valence can be scored as a whole, but these concepts can be extended by grouping words into sentiment categories around a predefined set of words such as *happy*, *sad* and *angry* [12].

Words grouped together because they share valence and because they are semantically related are known as affect categories. There are many ways of grouping words into affect categories. However, since the investigation of affect falls into the realm of psychology, this paper does not attempt to justify their use. This paper uses Ekman's categories, firstly because they were used in the paper on which this paper is based [10], but also because a shallow review of the literature indicated that Ekman's choice of categories are widely used and accepted. Ekman motivated their validity by offering evidence that they closely align with basic human facial expressions [12].

Ekman [12] mentions that all emotive words fall in one way or another into a basic set of six emotions consisting of ***fear***, ***anger***, ***enjoyment***, ***disgust***, ***sadness*** and ***surprise***. He points out that a range from pleasant to unpleasant is not sufficient to differentiate between differing emotions since the basic emotions differ in various aspects such as expression, behavioural response and other aspects. For this reason the focus on just positive and negative cannot describe the full picture. While there can be a multitude of words expressing anger there can be variations within the words in the fear family, so grouping the words into basic families this way does not dismiss the nuances of words within the emotive set. What it does do is organise them in a manner that makes them easier to compare and analyse [12].

4.1.2 Music terminology

Music theory can be described at many levels of complexity. The abstract mathematical aspects of the theory are completely avoided, and basic concepts are presented in a manner that attempts to present long known aspects of music theory in a way that can be understood by someone without knowledge of the domain. Section 4.3 presents trivial ways in which one could naively attempt to introduce emotive aspects into music. The current section would just like to explain the musical term "**mode**" as well as what is meant when discussing the terms "**major**", "**minor**" and "**triad**". No attempt is made to delve into the technical aspects of music literacy and the discussion is restricted to the diatonic scales found in Western music.

A scale can be started from any note. For example the scale of C can be played from the D to the D. A different choice of starting note alters the order of tones and semitones in the scale. There are seven modes or forms to a scale each starting on a different note. The altered order of tones and semitones is called a "**mode**". Modes are called "**major**" when the distance between the first and third notes equate to two tones and "**minor**" when the distance is one tone and a semitone. In modern music two of these modes, namely the "Ionian mode", which is one of the "**major**" modes and the "Aeolian mode", which is a "**minor**" mode are used more commonly than the others. Hullah [16] can be referred to for verification and further detail of this terminology.

A "**triad**" is a chord consisting of the tonic, third and fifth notes of the mode [27]. For the scale of C played from the note D the triad is composed of the notes D, F and A and is "**minor**".

4.2 Extracting the named entities

Natural Language Processing (NLP) is a large field within computer science. The location in the text of the various characters against which the melody is to be aligned is a task that needs to be done in order to achieve our research question. The task is achieved in as simple manner as possible without redirecting the scope of the paper in the direction of Named Entity Recognition (NER), which is a sub field within NLP. The task also does not attempt to analyse the best manner in which to extract affect categories from text and therefore uses the same mechanism used by Davis and Mohammad [10].

Project Gutenberg [2], which digitizes books whose copyright has expired, was chosen as a source of text of the novels. The books were chosen due to their popularity and to achieve a range of emotive choice.

The first task was to characterise the novels by counting the words in the section of text that fall into a particular affect category. The words of the text were looked up and classified using the NRC Word-Emotion Association Lexicon [1, 21].

A section of the text of the novel was analysed for it's top two affect categories as well as the top affect associated with the specific characters in the text. A small section was chosen instead of the full novel so that it was small enough to be able to identify correlation of characters with bars of music.

4.3 Introducing affect and valence into music

The introduction of emotion and effect into music is a complex cross domain exercise venturing into the realms of psychology. The literature is accessed for information that could contribute to making the characteristics between characters stand out from one another. However, the search is shallow and no attempt is made to verify or assess the accuracy of the information.

Some features of music such as the choice of major or minor modes produce far more unambiguous and reliable expression of a particular affect than others with mode being the most important parameter determining valence [15, 31, 6]. The major mode is very strongly associated with gaiety, playfulness, happiness and sprightliness, and the minor mode with sadness, sentimental yearning and tenderness [15]. Triads form the base of the emotional response to music [6]. Minor triads are slightly more ambiguous in conveying meaning than major triads [5]. When extending beyond the major and minor modes the modes in order from the most negative valence to the most positive are: ***Locrian***, ***Phrygian***, ***Aeolian***, ***Dorian***, ***Mixolydian***, ***Ionian*** and ***Lydian*** [31]. This is partly due to the fact that the former modes have a flattened third degree and are therefore considered to be minor scales [31].

Negative valence melodies are centered around C4 (middle C on a piano) and more positive valence two octave above that [31]. Compositions written in the major mode are not always merry and playful they are just more playful than the same compositions written in minor modes [15]. Dissonant harmonies are known to be exciting, agitating, vigorous, and sometimes sad while simple consonant harmonies are happy, graceful and serene, but the reliability of the produced affect is not as guaranteed as it is with mode [15].

Cohen [5], in discussing the known aspects of film soundtrack that contribute affective meaning, mood and tonalities in film narrative, concurs that high tones produce happier sounding music than low, and fast tones produce happier sounding melodies than slow, but points out that combinations of these aspects, such as high tones played slowly, are less predictable. She also points out that ascending or descending major triads, are more strongly associated with happiness than repeated notes. The direction of melody is also of much less importance than the harmony or rhythm of the music [15]. Melody is also much more complex than mode, making it more difficult to assess its affective contribution [15]. A melody can ascend slowly or rapidly in small leaps or giant skips. Certain aspects of the melody can be assessed independently.

A composition with a firm rhythm will be much more serious, dignified and vigorous than the same composition with a flowing rhythm [15]. Holding the harmony and rhythm constant while altering the melody doesn't predictably change the affect [15].

Certain dissonant chord structures can also impart specific affect. Augmented chords, which are triads consisting of two major thirds, which do not occur as part of the western scale, make the music more exciting and impetuous, and diminished and minor seventh chords suggest depression or yearning [15]. In addition a smaller pitch space corresponds to more positive valence [32].

4.4 Methods and techniques of computational music generation

This section surveys the existing computational music generation literature to assess techniques that have been successfully used in the generation of music.

Due to the hierarchical nature of music, it lends itself to being represented by a Context Free Grammar (CFG) [18]. There is more than one choice of possible grammar and the rules can be chosen to be very general or very specific for a given style of music [23]. Probabilities for CFGs, matching a particular style of music, can also be learned from existing music [13] and stochastic elements can be introduced. N-grams representing the probabilities of the choice of production rule based on the previous $n-1$ choices can then be learned from existing pieces of music [13], further increasing the choice of suitably chosen production rules. Thus, Bayesian reasoning also takes its place in music generation along with related techniques such as entropy calculation in the choice of probabilities [29].

An issue with generating music this way is that while music has a set of predetermined rules, part of the novelty in music is the aspect of freshness and surprise and it is therefore not uncommon for the rules to be intentionally broken by musicians.

Another way of varying musical phases that makes use of music's hierarchical nature is the practise of building and then mutating tree structures. Starting at a random node and randomly collapsing branches eliminates notes or phrases [30].

The expectation (or expected value) of music tends to resolve [24]. This applies both from an intuitive point of view whereby phrases with dissonance or less common notes in the scale will tend to end on more pleasing, less dissonant and more commonly used notes, such as the first, third or fifth of the scale and also toward the statistical interpretation of expectation where the expected values of notes at the end of a phrase would be in the averaged, commonly used notes. It is common for a melody to set up expectation of the following note of the melody by setting up a wider interval between notes and then giving the listener a sense of resolve and closure by changing the direction of the melody with a smaller interval. Tension can be induced by violating this expectation. There are also statistical regularities in the size and direction of melodic intervals. It has been pointed out that music has meaning and expressive effect specifically when expectation is violated in some way [29]. Sometimes these expectations come from standard musical theory, for example harmonic cadences, in which some chord patterns are more common than others [31]. If these expectations are not satisfied, listeners can experience a sense of surprise. Musical *surprise* can then be quantified by how far it is from the expectation.

Another common way of generating music comes from the recognition of the long known relationship between music and mathematics. A genre known as algorithmic music uses various mathematical functions, including fractals, to produce music [18, 11].

Regardless of the model, most methods of music generation are combined with some form of evolutionary method of generation and many of the models and methods overlap [23]. Chance based, rule based and probability based algorithms have been used, with some researchers using more than one type within the same composition [4]. The reason is the different strengths and weaknesses of each. For example, stochastic methods tend to evolve too much to keep the music sounding like the same song, and they lose the fairly structured overall harmonic form of the composition. Grammars and Markov Chains are good at high level structure and probability methods perform well at note choice based on distributions [4].

While no particular algorithm has yet gained preference over any other [4], Genetic Algorithms (GAs) appear to be commonly used in the current computational music research. One of the reasons is because music composition has a very large search space and genetic algorithms are known to be very good at dealing with this problem [22]. Evolutionary algorithms (EAs) are advantageous in composition since they don't depend on human creativity for generating music [23].

Affect	Mode	Affect	Mode
Fear	Phrygian	Sad	Aeolian
Anger	Dorian	Disgust	Mixolydian
Joy	Ionian	Surprise	Lydian

Table 1: Correspondence of affect to musical mode as used by the fitness calculation

5. METHOD

JMusic [20] was used for music representation and processing and JGAP [19] was used for the GA.

The pitch and duration of the first and second melody was modelled and the top affect was introduced into the first melody and the second highest affect into the second melody line using the fitness function.

Existing MIDI files, which were chosen so that the affect categories were represented, were used as input in the algorithm. This was done by choosing music from films where the plot indicates the intended affect or by choosing songs where the title indicated the affect.

Unlike the research by [10], the implementation of affect in the melody was not produced by predetermined rules, such as those for tempo and pitch. The affect was instead modelled using features of the input melody and by the choice of fitness function.

A MIDI file for the specific affect intended for the current melody line was randomly chosen from the songs that have been identified with the intended affect and used as input for the melody. Specifically the tempo of the first MIDI file chosen for the first melody line was used as the tempo for the output song and the range of notes in the MIDI file contributes towards the fitness function.

Statistical attributes of the input melody were compared with those of the generated melody and the divergence between the two was appropriately scored. These statistical attributes included semitone count, large jumps count, melodic direction stability[1] and consecutive identical pitch count. These particular statistics were used because they are built into JMusic and were therefore readily available.

Avoiding large melodic intervals is one of the core rules featured in almost every academic work on voice-leading [17] and occurrences were penalised by deducting increasingly larger values from the fitness score as the intervals increased.

The easiest way to recognise different characters melodies would be to assign each a different instrument, and leave the notes out when the characters don't appear. However, this contributes very little towards investigating the introduction of the affect of characters into a melody line and is not very interesting. All melodies therefore use the same instrument.

Use is made of the statement that the modes in order from the most negative valence to the most positive are: ***Locrian***, ***Phrygian***, ***Aeolian***, ***Dorian***, ***Mixolydian***, ***Ionian*** and ***Lydian*** [31]. The *Locrian mode* was discarded, because it has an unusual chord at the first note of the scale (the diminished chord).

By assuming that the Ekman categories can be ordered from most negative to most positive as: ***Fear***, ***Sad***, ***Anger***, ***Disgust***, ***Joy*** and ***Surprise***, a one to one mapping is achieved. Clearly this is an extremely naive assumption and simplification, but it is sufficient for the purposes of separating the affect of the different characters.

[1]the ratio between the number of consecutive pitch steps in the same direction and the total number of notes

This mapping of valence to mode was used by the GA's fitness function as described in the results. The mapping is shown in Table 1.

6. RESULTS

6.1 Text extraction

The results of data mining for the affect as per the NRC Emotion Lexicon over the entire novel seemed contradictory, which makes sense since many emotions can occur from chapter to chapter. For example *Hamlet* and *Romeo and Juliet*, which are tragedies had joy as their second and first highest emotion respectively and scored as being more positive than negative. The figures are shown in Table 2. The results also imply that some of these affects may be more common than others. In particular, Table 2 appears to indicate that *fear*, which was the top affect for nine of the twelve novels, seems a little too highly represented.

Since any storyline will have chapters with differing affect, which was inferred by the sometimes contradictory contrast between the highest and second highest affect categories, and since these were too high level for the character affect that was needed, a representative portion of chapters from a few of the novels was chosen and the analysis was redone on the smaller section of text. Non neutral words are the total count of words that were tagged by the lexicon as being either positive or negative.

Trust was the highest affect calculated in *Peter Pan* and *Alice in Wonderland* according to the NRC Word-Emotion Association Lexicon [1, 21], in agreement with [10]. *Trust* as an effect was disgarded for the purposes of this research because only the six Ekman emotions were taken into account. Other values were not in agreement. For instance, *trust* would have been the highest affect for *Heart of Darkness* because it had the highest wordcount of 935 words. *Heart of Darkness* also contained more positively tagged words than negative.

Initially, an attempt was made at NER using Apache's OpenNLP library [3]. It had a hard time with the Shakespeare names and wasn't distinguishing between place names and people. In addition, determining that *Hamlet* and *Lord Hamlet* referred to the same entity and associating pronouns with a particular character was also a problem. As mentioned in project limitations, this was not intended to be an extensive NLP task and therefore, the identification of characters was done manually, and because it was very time consuming, the manual recognition was only done for one book, *War of the Worlds*. Since the text processing was not automated, there was no reliable way to tell which valence words, and hence counts, belong to which characters. The task was simplified by associating the affect words co-ocurring in the paragraph in which the character appears.

For simplicity a chapter of the book was broken down into paragraphs with the assumption that each paragraph would be represented by one bar of music. Table 3, shows the paragraph number, the first few words of the paragraph, and which characters occur in the paragraph. Each paragraph was then measured for affect, by looking up each word in the paragraph. The resulting counts of how many words were from each affect category are shown in Table 3.

The affect of the specific characters was then calculated in the following simple way from Table 3.

The words in all of the paragraphs in which the character appears were grouped into Ekman categories. The Ekman category

Book	total words	non-neutral words	emotive words	1st affect	2nd affect	valence
Dracula	82929	11873	14562	fear : 3357	joy : 2858	positive
The War of the Worlds	33060	4662	6901	fear : 1388	sad : 1215	negative
Peter Pan	27808	3425	5178	joy : 889	sad : 842	positive
Pinocchio	23361	3160	4819	joy : 842	sad : 715	positive
Heart of Darkness	21774	3380	4995	fear : 925	sad : 885	positive
Hamlet	19222	3400	3895	fear: 762	joy: 753	positive
Romeo and Juliet	17172	2892	3759	joy : 838	sad : 763	positive
Alice in Wonderland	15999	1602	1801	fear : 360	joy : 353	positive

Table 2: Summary of word counts and affect word counts in various novels with the resulting highest and second highest affect as per the NRC Emotion Lexicon

	Paragraph starting Text	Characters	highest affect	secondary affect	valence
All	total words: 1453 emotive words: 266 non neutral words: 195		fear: 78	sad: 57	negative
1	It was while the curate had sat and talked so wildly to me ...	narrator, curate, Martians	joy: 2	anger, fear, disgust: 1	positive
2	But three certainly came out about eight o'clock and, advancing slowly ...	Martians	joy, fear, surprise: 2		positive
3	It was this howling and firing of the guns ...	Martians	fear, surprise: 3		negative
4	The St. George's Hill men, however, were better led or of a better mettle.	men, Martians	joy, sad, fear, surprise: 1		negative
5	The shells flashed all round him, and he was seen to advance a few paces, stagger, and go down.	Martian	fear: 6	surprise: 3	negative

Table 3: War of the Worlds. Characters by paragraph for the first few paragraphs of chapter 15

of the group containing the most words was chosen as that characters' affect. This manner of assigning affect to characters was very simplistic and results in some words contributing to the affect of more that one character. The valence was calculated in the same manner.

As an example of this calculation, consider paragraph one. All three characters score two towards the valence, *joy*. The only characters appearing in paragraph two are the Martians. They were therefore assigned a two in the *fear* and *surprise* valences and the existing valence, *joy* was incremented by two.

Whether the character's valence was more positive or negative, was calculated similarly by summing the valence counts of the paragraphs in which they appear. The results of this calculation are presented in Table 4. The curate only appears in two paragraphs and pronouns such as "us" and "them" were not taken into account.

6.2 Music generation

The GA was a straight forward representation of pitch and note duration as a bound integer range and float range. The fitness function was a sum of musical features and features that were added to take care of disssonance between melodies. It was faster for the algorithm to not convert between the gene representation for the algorithm and JMusic, since JMusic slowed the learning rate drastically.

The first two melody lines were generated first. The MIDI files that had been categorised by affect were used to produce some of the affect. Each melody was randomly assigned one of the MIDI songs that had been chosen as belonging to the intended affect. The tempo of the song chosen for the first melody line was used as the tempo for the resulting score. All melodies were assumed to be in the key of C. The resulting tempos were consistent with the remarks in the literature, with ***sad*** tunes being slower than the more upbeat affects of ***joy*** and ***surprise***. The rest of the affect was influenced through the fitness function.

6.2.1 Fitness function

Fitness functions for generating music are difficult to design [4, 33]. There are three types commonly used, namely knowledge based, human user and neural network. A weighted sum of compositional features is common [23] and this is what is used here.

The weighted sum of compositional features is very successful even when only a few features are part of that calculation. The more known music rules that are added as features the more melodic the music became which was expected.

The fitness function was knowledge based and contained features for calculating score as per the weighted sum:-

$$\begin{aligned} Score = {} & w_0 C_l(chr) + w_1 D(chr, midi) + \\ & w_2 M(chr, aff) + w_3 P(chr) + \\ & w_4 C_h(chr, bars) + w_5 V_l(chr) + \\ & w_6 S(chr, midi) \end{aligned}$$

where the weights w_0, w_0, ..., w_6 were left as one, so that the fitness score features counted equally, and the $C_l(chr)$, $D(chr, midi)$, $M(chr, aff)$, $P(chr, val)$, $C_h(chr, bars)$, $V_l(chr)$ and $S(chr, midi)$ were functions that took the GA's chromosome (*chr*) and the MIDI file features that had been calculated for each part's MIDI file (*midi*), the intended affect (*aff*), intended valence (*val*) and the bar numbers (*bars*) of the character appearances, and calculated a score based on the attributes enumerated in Table 5. Pseudocode for $C_l(chr)$, $D(chr, midi)$, $M(chr, aff)$, $P(chr, val)$, $C_h(chr, bars)$, $V_l(chr)$ and $S(chr, midi)$ can be found in Tables 6, 7, 8, 9, 10, 11 and 12 respectively.

Of these feature scoring functions, $C_l(chr)$, $D(chr, midi)$, and $V_l(chr)$ are encouraging basic known musical rules.

The $M(chr, aff)$, $P(chr, val)$ and $C_h(chr, bars)$ fitness rules were the ones encouraging affect. $P(chr)$ is also mostly driven by basic rules of music, since it is known that in the key of C, the more sharps and flats exist in the melody the more dissonant it is and therefore the more negative. $C_h(chr, bars)$ is also fairly straightforward in that it encouraged a characters' melody line to be silent in those bars that didn't coincide with the characters' appearance. The $M(chr, aff)$ feature scoring function was the main fitness function driving the intended affect of the melody and requires more explanation.

	Purpose
$C_l(chr)$	Encourage close notes by making sure that the resulting note pitches of the melody lines didn't make large leaps.
$D(chr, midi)$	Made sure that the duration of the notes had similar length to the input MIDI tune assigned to each part
$M(chr, aff)$	Encouraged the melody to adhere to the mode that mapped to the affect category for that melody line. It also encouraged adherence to the chosen valence through the encouragement of major or minor.
$P(chr, val)$	If the valence of the melody is required to be negative it encourages more dissonance by allowing more sharps and flats. For a positive melody it encourages adherence to the natural notes
$C_h(chr, bars)$	When aligning a melody to the appearance of a character in a particular bar of music, this fitness scorer produces a lower score for notes that were not rests that exist in *bars* that the character shouldn't appear in. This scorer was discarded when the character's melody was generated using phrases from the MIDI file
$V_l(chr)$	Encouraged the notes between melodies to produce pleasant intervals when the notes co-occur.
$S(chr, midi)$	Encouraged similar characteristics in the generated tune to those calculated from the MIDI provided melody. For example if the MIDI file and the GA generated melody had a semitone count within a small range of each other then the score was increased

Table 5: Summary of fitness functions used by the GA for music generation

As discussed in the section of sentiment in music 4.3, it is well known that the mode of a piece of music is critical to its affective state.

The naive mapping of modes to affect as described in Section 5 is part of the $M(chr, aff)$ fitness calculation. The fitness scoring function was encouraged to produce higher scores if the GA had produced melody containing the notes at the first, third and fifth of the mode. This way the melody was encouraged to stay in the mode that had been assigned to it. This worked surprisingly well. Examples of the output with audio, can be found online at [8].

Precondition: *chr* is the current chromosome representing the melody lines that has been produced by the GA
Postcondition: Score is deducted for chromosomes representing melody that contains large leaps between notes

```
function C_l(chr)
    convert chr to note pitches
    score ← 0
    for each two consecutive notes do
        span ← absolute distance between notes in semitones
        if span > 5 then                          ▷ Perfect fourth interval
            decrease score a large amount
        else
            if span > 4 then                      ▷ Major 3rd interval
                decrease score a little bit
            else
                increase score
            end if
        end if
    end for
    return score
end function
```

Table 6: Fitness function ensuring melody doesn't make large leaps

Precondition: *chr* is the current chromosome representing the melody line and *midi* is the statistics calculated from the provided MIDI file that was randomly assigned to this melody line because it matched the intended affect
Postcondition: Score is deducted or increased depending on whether generated note durations match those in the MIDI file

```
function D(chr, midi)
    convert chr to note duration
    midiDurations ← durations of the notes in the MIDI file
    score ← 0
    for each note's duration do
        if midiDurations contains duration then
            increase score
        else
            decrease score
        end if
    end for
    return score
end function
```

Table 7: Fitness function ensuring generated notes prefer a duration in line with the MIDI file assigned to that melody

Precondition: *chr* is the current chromosome representing the melody and *aff* is the intended affect
Postcondition: Score is deducted when the mode mapped to the affect is in disagreement with the generated notes

```
function M(chr, aff)
    convert chr to note pitches and note durations
    score ← 0
    mode ← Mode as specified by the one-to-one mapping to affect
    isPassingNote ← True if note duration is very short and does not belong to
the mode
    for each note do
        isInMode ← True if note pitch belongs in mode
        if note equals tonic or median of mode then
            increase the score by a large amount
        else if note is the fifth of mode then
            increase the score a small amount
        else
            decrease the score
        end if
        if isPassingNote and not isInMode then
            decrease the score
        end if
    end for
    return score
end function
```

Table 8: Fitness function encouraging adherence to mode by rewarding notes that match the tonic triad of the mode and penalising passing notes that do not belong to the mode

Character	Paragraphs where they appear	Accumulated scores	Adopted Affect	Adopted Valence
narrator	1, 9, 15-22	fear:13, sad:5, surpise:4, joy:3, disgust:0, anger:0, positive:4, negative:6	fear	negative
curate	9, 16	joy:2, fear:0, surpise:0, disgust:0, anger:0, sad:0, positive:2, negative:0	joy	positive
Martians	1-8, 10-12, 14, 18, 19, 22, 26, 27, 29-32	fear:37, surpise:12, anger:12, sad:12, joy:7, disgust:3, positive:5, negative:16	fear	negative

Table 4: The affect of specific characters appearing in chapter 15 and the resulting highest affect choices for the specific characters

Precondition: *chr* is the current chromosome representing the melody and *val* is a boolean indicating whether the valence is intended to be positive
Postcondition: Score is deducted when the melody valence is intended to be positive but the melody is dissonant

```
function P(chr, val)
    convert chr to note pitches
    score ← 0
    for each note pitch do
        if val is true then                          ▷ Valence is positive
            if note is a sharp or flat then
                increase score
            else
                decrease score
            end if
        else                                         ▷ Valence is negative
            if note is a sharp or flat then
                decrease score
            else
                increase score
            end if
        end if
    end for
    return score
end function
```

Table 9: Fitness function rewarding greater dissonance when the valence intent is negative and more reward for less dissonance for a positive valence. Since we have assumed a key of C this equates to rewarding a greater number of sharps and flats for a negative valence.

Precondition: *chr* is the current chromosome representing the melody and *bars* are the bar numbers in which the character is expected to appear
Postcondition: Score is deducted when the character's melody occurs in a bar that does coincide with a character in the paragraph

```
function C_h(chr, bars)
    score ← 0
    if bars is empty then                            ▷ Not a character melody
        return score
    end if
    convert chr to note pitches and duration
    for each note do
        barOfCurrentNote ← Calculate which bar this note is in
        if bars contains barOfNote then
            increase score
        else
            decrease score
        end if
    end for
    return score
end function
```

Table 10: Fitness function that encourages rests in bars of music that should coincide with a character from the novel.

Precondition: *chr* is the current chromosome for all melodies
Postcondition: Score is deducted when two melody lines clash with each other

```
function V_l(chr)
    score ← 0
    convert chr to note pitches and durations
    for notes between melodies that occur at the same time do
        interval ← distance in semitones between notes
        if interval is octave then
            increase score by a medium amount
        else if interval is a unison then
            decrease score by a small amount
        else if interval is consonant then ▷ Perfect fourth/fifth, major/minor third
            increase score by a large amount
        else if interval is dissonant then           ▷ major/minor second, ninth
            decrease score by a large amount
        else
            score unchanged
        end if
    end for
    return δ
end function
```

Table 11: Fitness function rewarding intervals between notes between melodies that are less dissonant

Precondition: *chr* is the current chromosome representing the melody lines that has been produced by the GA and *midi* is the statistics calculated from the provided MIDI file
Postcondition: Score is deducted when the generated melody line has statistics that differ from the MIDI file that has been assigned to it

```
function S(chr, midi)
    convert chr to note pitches and durations
    score ← 0
    threshold ← magnitude that must be exceeded to be considered divergent
    gastats ← Calculate statistics of generated melody
    for each statistic in gastats do
        if gastats > threshold then
            decrease score
        else
            increase score
        end if
    end for
    return score
end function
```

Table 12: Fitness function ensuring melody has similar statistical features to the provided MIDI file

Initially the melodies associated with a character were generated in the same manner as the first two melody lines. It was impossible to distinguish between the bars of music in which the character was supposed to be identifiable and bars in the music which did not coincide with the character. Adjusting the fitness function to encourage silence in the bars of music which are associated with paragraphs in the novel in which the characters did not appear did not fix the problem.

An attempt was then made to adopt some of the techniques from film music that allocate a specific melody or phrase to a character. This unfortunately resulted in the GA being abandoned for the remaining melody lines since it was not achieving the synchronisation with the characters that was required, using the fitness functions as they had been designed.

A brief foray into the music literature, some of which was discussed in Section 4.3, turned up surprisingly few results concerning the technical aspects and implementation of film music techniques. However, a demonstration complete with musical examples is discussed and demonstrated by [9] in the featurette discussing the creation of the music for the TV series, "Eureka". Cosby and Paglia [9] demonstrate how music soundtracks frequently coincide with characters within the visual. It is quite common for a character to have a specific phrase (or instrument) in the musical theme coincide with their appearance throughout the movie [14]. This is known as a leitmotif. The phrase for that character changes only in minor ways. It is played more slowly, or at a slightly different pitch.

All melodies, including those for characters already had a random MIDI file assigned to them that was from those tagged with their affect category. From that MIDI file a random phrase of up to six notes was chosen and assigned to the character. The phrase length was chosen arbitrarily as this is a reasonable length for a phrase of music. The music generated for a paragraph is laid out according to the following conventions. Two melody lines provide the overall for a paragraph. Below these two lines, a melody line is provided for each character relevant to the paragraph. Each such character melody line is generated as a random mutation of the character's chosen phrase. The ten allowed mutations, conveniently provided by JMusic are shown in Table 13.

The figures that follow show the resulting program output.

The resulting audio files and complete scores can be accessed and listened to at [7]. The third stave from the top contains the Narrator's melody, the fourth stave contains the Martians' melody and the bottom stave depicts the melody belonging to the Curate.

For the *War of the worlds* melodies, the top melody line was generated for the highest affect of the novel, which, according to the results in Table 4, was *fear*. Similarly the second melody, which is the second from the top was generated for the affect *sad*. The valence of the novel was negative and this was used for both of these melody lines.

The results from Table 4 then determined the following for the character melodies: The third melody, which is in the third stave from the top, coincides with *the narrator* whose melody required the affect *fear* and required a negative valence. Similarly, melody four is *the Martians* with the affect *fear* and a negative valence. Lastly, the bottom melody represents *the curate* with an affect of *joy* and a positive valence.

One expects the melody for the curate to remain silent until his expected entry at bar sixteen and, not unexpectedly, his melody reappears slightly early in bar fifteen as is shown in Figure 2 which consists of bars thirteen to seventeen. A variation of the narrator's theme presented itself on time in bars fifteen, sixteen and seventeen.

Name	Description
Shuffle	Randomise the order of notes in the phrase without repeating any note
Vary length	Varies all Note duration randomly between specified values
Transpose	Transpose all the notes a random distance of one to five notes with in the confines of the characters mode
Add to length	Vary both the rhythm value and duration of each note in the phrase by a random amount between 0.0 and 1.0
Add to rhythm	Vary the rhythm value of each note by a random amount between 0.0 and 1.0
Rotate	Move the notes around the phrase, first becoming second, second becoming third, etc. with last becoming first
Increase duration	Vary the duration of each note by a random multiplication factor between 0.0 and 1.0
Add to duration	Vary the duration of each note by a random amount between 0.0 and 1.0
Do nothing	Use the phrase of notes as is without alteration
Change length	Reconstructs the phrase so that it's notes are stretched or compressed to the length specified

Table 13: Mutations of the character's phrase that are randomly added at suitable places in their melody line

The first four bars of the example melody and the discussion below refer to Figure 1a. The narrator's generated phrase has been highlighted in purple and consists of one sustained note which is then repeated once on it's own and once again with a second note, both of which are then continued for some duration. In musical terms this is called a sustain and is indicated by the curved horizontal line between notes that are the same. In music notation this horizontal line is known as a tie. The note at the end of the tie is not played again. The narrator doesn't appear in paragraph (and therefore bar) two but the phrase manipulation that extends the duration of the notes has slightly extended it into the second bar. The Martian phrase is highlighted in blue and consists of a four note phrase which is ascending. As this character was in all of the first eight bars the phrase is rapidly repeated in the second, third and fourth bar with minor variations as well as in isolation in the following four bars which are not shown here. The narrator has one or two random notes that occur over this time but his phrase does not repeat.

Bars nine to twelve are shown in Figure 1b. It can be seen that the narrator's phrase reappears slightly before his expected entry point in bar 10. The curate also makes his first appearance on time in bar nine. His generated melody consists of a note that repeats once and is highlighted in green. The Martian melody which should occur in bars eight and ten has extended slightly into the ninth bar. The four short notes of the Martian melody are quite distinct from the sustained melody of the narrator and both are distinct from the simple two note phrase of the curate. The mutations that allow the phrases to extend slightly beyond their expected bars could probably have be made shorter or been disallowed, but the slight "drifting" of the melody that this causes doesn't really distract from the understanding of occurrence of characters. It may even provide continuity

(a) Bars 1 to 2 (above)
and 3 to 4 (below)

(b) Bars 9 to 10 (above)
and 11 to 12 (below)

Figure 1: Generated output for “War of the worlds” showing two melody lines for affect and one for each character

Figure 2: Bars 13 to 14 (above)
and 15 to 17 (below)

and expectation of their arrival. It could be interesting to explore slightly longer phrases and additional work needs to be done to align the character melodies with the harmony of the non character lines. The remaining bars of the melody continued to be consistent with the ones shown here and can be downloaded at [7].

Attempts to acknowledge the repetitive elements and musical form that help keep pieces of music cohesive and offer the listener familiarity, would also enhance the generated result.

7. CONCLUSION

The visual and audio output indicates that music generated by a computer can introduce a melody line that identifies the presence of particular characters at particular points in the music. The generated music can indicate the intended affect of the character. In addition characters appearing simultaneously can still be recognised even if they both have the same intended affect category assigned to each of them.

What is less clear is how repetitive the characters’ musical phrases are required to be in order for them to remain discernible and whether GAs or any other EA is capable of producing distinguishable leitmotifs.

8. ACKNOWLEDGEMENTS

Thanks need to be extended to Adele Botha at the Meraka Institute, for her helpful comments, particularly with reference to general structure and suggested improvements.

References

[1] NRC Word-Emotion Association Lexicon (Version 0.92). Available: http://www.saifmohammad.com/WebPages/lexicons.html, 2011.

[2] Project Gutenberg. Available: http://www.gutenberg.org, 2015. (Visited on: 2014-06-11).

[3] Apache Software Foundation. Apache OpenNLP. Available: https://opennlp.apache.org, 2015. (Visited on: 2015-02-19).

[4] D. Cerny, J. Suthar, and I. Geselowitz. U-AMP: User input based algorithmic music platform. 2013.

[5] A. J. Cohen. Associationism and musical soundtrack phenomena. *Contemporary music review*, 9(1-2):163–178, 1993.

[6] N. D. Cook. *Tone of Voice and Mind: The connections between intonation, emotion, cognition and consciousness*. Advances in Consciousness Research. John Benjamins Publishing Company, 2002.

[7] Coral Featherstone. Character recognition by character for chapter 15 of War of the worlds. Available: https://diminishedtonic.wordpress.com/2015/01/13/character-recognition-by-character-for-chapter-15-of-war-of-the-worlds, 2015. (Visited on: 2015-01-13).

[8] Coral Featherstone. Computationally generated music for the basic Ekman emotions. Available: https://diminishedtonic.wordpress.com/2015/01/13/computationally-generated-music-for-the-basic-ekman-emotions, 2015. (Visited on: 2015-01-13).

[9] A. Cosby and J. Paglia. Eureka: Season 3.0, Creating a musical personality featurette, 2008. Produced by NBC Universal Television and Universal Cable Productions.

[10] H. Davis and S. Mohammad. Generating music from literature. In *Proceedings of the 3rd Workshop on Computational Linguistics for Literature (CLFL)*, pages 1–10, Gothenburg, Sweden, 4 2014. Association for Computational Linguistics.

[11] G. Diaz-Jerez. Algorithmic music: using mathematical models in music composition. *The Manhattan School of Music*, 2000.

[12] P. Ekman. An argument for basic emotions. *Cognition & Emotion*, 6(3-4):169–200, 1992.

[13] J. Gillick, K. Tang, and R. M. Keller. Machine learning of jazz grammars. *Computer Music Journal*, 34:56–66, 2010.

[14] C. Gorbman. Narrative film music. *Yale French Studies*, pages pp. 183–203, 1980.

[15] K. Hevner. Experimental studies of the elements of expression in music. *The American Journal of Psychology*, pages 246–268, 1936.

[16] J. Hullah. *Rudiments of musical grammar, the substance of lectures*. YOYO Media, 2013. A replication of a book originally published in 1857. Available: http://books.google.co.za/books?id=1DQDAAAAQAAJ.

[17] D. Huron. Tone and voice: A derivation of the rules of voice-leading from perceptual principles. *Music Perception*, 19(1):1–64, 2001.

[18] H. Järveläinen. Algorithmic musical composition. 2000.

[19] JGAP. Java Genetic Algorithms Package. Available: http://jgap.sourceforge.net/, 2015. (Visited on: 2015-01-08).

[20] JMusic. Computer music composition in Java. Available: http://explodingart.com/jmusic, 2003. (Visited on: 2014-09-15).

[21] S. M. Mohammad and P. D. Turney. Crowdsourcing a word–emotion association lexicon. *Computational Intelligence*, 29(3):436–465, 2013.

[22] G. Papadopoulos and G. Wiggins. A genetic algorithm for the generation of jazz melodies. *Proceedings of STEP*, 98, 1998.

[23] G. Papadopoulos and G. Wiggins. AI methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB Symposium on Musical Creativity*, pages 110–117. Edinburgh, UK, 1999.

[24] M. T. Pearce and G. A. Wiggins. Expectation in melody: The influence of context and learning. 2006.

[25] K. Peffers, T. Tuunanen, C. E. Gengler, M. Rossi, W. Hui, V. Virtanen, and J. Bragge. The design science research process: a model for producing and presenting information systems research. In *Proceedings of the first international conference on design science research in information systems and technology (DESRIST 2006)*, pages 83–106, 2006.

[26] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. A design science research methodology for information systems research. *Journal of management information systems*, 24(3):45–77, 2007.

[27] W. Porter and L. Mason. *The Musical Cyclopedia: Or, The Principles of Music Considered as a Science and an Art: Embracing a Complete Musical Dictionary, and the Outlines of a Musical Grammar, and of the Theory of Sounds and Laws of Harmony, with Directions for the Practice of Vocal and Instrumental Music, and a Description of Musical Instruments*. J. Loring, 1834. Available: http://books.google.co.za/books?id=mQ4oAAAAYAAJ.

[28] C. Strapparava and A. Valitutti. Wordnet affect: an affective extension of wordnet. In *LREC*, volume 4, pages 1083–1086, 2004.

[29] D. Temperley. *Music and probability*. The MIT Press, 2007.

[30] B. Thom. Artificial intelligence and real-time interactive improvisation. In *Proceedings from the AAAI-2000 Music and AI Workshop*, pages 35–39. Citeseer, 2000.

[31] I. Wallis, T. Ingalls, E. Campana, and J. Goodman. A rule-based generative music system controlled by desired valence and arousal. In *Proceedings of 8th international sound and music computing conference (SMC)*, 2011.

[32] G. Widmer and W. Goebl. Computational models of expressive music performance: The state of the art. *Journal of New Music Research*, 33(3):203–216, 2004.

[33] G. A. Wiggins, G. Papadopoulos, S. Phon-Amnuaisuk, and A. Tuson. Evolutionary methods for musical composition. *DAI RESEARCH PAPER*, 1998.