MUSE prototype for Music Sentiment Expression

Ralph Abboud and Joe Tekli
E.C.E. Department, School of Engineering
Lebanese American University
36 Byblos, Lebanon
ralph.abboud01@lau.edu, joe.tekli@lau.edu.lb

***Abstract*—This paper briefly describes and evaluates *MUSE*, a MUsical Sentiment Expression prototype system, taking as input a MIDI music file and producing as output a sentiment vector describing the 6 primary emotions (i.e., anger, fear, joy, love, sadness, and surprise) expressed by the music file.**

***Keywords*—Music Analysis, MIDI, Sentiment Analysis, Supervised Learning, Fuzzy K-Nearest Neighbors.**

I. INTRODUCTION

Over the past few years, text-based sentiment analysis tools have evolved into mature services and APIs. For instance, tools like LIWC (Linguistic Inquiry – Word Count) [1] and IBM's ToneAnalyzer [2] can extract sentiments from texts to report and predict expected user feedback However, very few comparable breakthroughs have been made when it comes to analyzing multimedia documents (e.g., images, sounds, and videos).

Musical Sentiment Analysis (MSA) attempts to bridge this gap between text and music. Given an input musical piece, an MSA tool should accurately estimate end users' emotional response when listening to the given piece. The potential applications of such a sentiment analysis system are broad and could have a serious impact in the field. For one, it could help music producers gauge their compositions to check whether they will produce the target sentiments they were attempting to portray. Beyond that, it could usher in a new sentiment-based music search functionality, in which musical pieces are retrieved based on their expected sentiment vectors. Most importantly, it could herald the start of the development of a universal retrieval system, where any multimedia document of any type (including images, videos, music, etc.) could be retrieved based on its perceived sentiment vector, irrespective of the media-specific features (e.g., visual, musical, spectral) that are part of its nature, and which are only dealt with at the sentiment-analysis stage.

In this paper, we concisely describe our musical sentiment analysis prototype system titled *MUSE* (MUsic Sentiment Extraction), which predicts a user's emotional response when listening to a given symbolic musical piece (presented in MIDI[1] format) [3]. The remainder of this short paper is organized as follows. Section II briefly presents related works. Section III highlights system constraints. Section IV describes the overall system architecture and main components. Section V describes our experimental evaluation, before concluding in Section VI.

II. RELATED WORKS

Music Sentiment Analysis (or MSA) is one of many open problems within the broader field of Music Information Retrieval (MIR), which deals with the representation, description, storage, and retrieval of information from music. Much like with standard IR systems, MIR systems (and MSA systems in particular), convert music documents into feature representations. These range over *high-level symbolic* features (a.k.a.[2] *music-theoretic* features, based on musical note abstractions, such as musical key and chord progressions) and *low-level frequency-domain* features (a.k.a. *statistical* features, based on frequency data used to describe audio formats, such as spectral components of audio samples and frequency histograms) [4]. Many approaches in the literature combine both these feature ranges into so-called multimodal feature vectors [5]. Some approaches in MIR have also built on breakthroughs in text-based sentiment analysis to improve musical sentiment analysis, by incorporating music lyrics into the repository entries to be analyzed [6].

However, MSA research has not always gone in that direction. In fact, one of the earliest MSA solutions, developed in the late 1980s by Katayose *et al.* [7] firmly placed its emphasis on purely musical features. In this approach, the authors develop an artificial music expert, a system that can detect and treat music just like any human intuitively does: through its emotions. To do this, they introduced "*quasi-sentiments*", a semantic/emotional meaning behind a given piece, so as to emulate how a human would react to a piece. Their extraction technique consists of mapping musical phenomena to these quasi-sentiments using a set of pre-defined rules. For example, a certain chord progression could correspond to a gloomy emotion, while a certain key or tempo could indicate a happy emotion. Through a simple rule-based approach, the authors were able to use musical features extracted from the input musical piece to infer its underlying emotions.

More recent efforts attempt to use as many features as possible, be it content-based (symbolic and/or sampled audio) or textual (lyrics of a song) to extract the sentiments of a given musical piece [6] [8] [9] [10]. For example, Panda et.al [6] perform sentiment-based retrieval based on a set of 253 simple musical features, 98 melodic features, 278 symbolic audio features, and 19 lyrical features. From this very large feature set, the authors seek to select the best combination of features to perform the sentiment analysis task. Results, based on optimal feature selection and retrieval performance testing for multiple machine learning and classification algorithms (SVM[3], k-NN[4], etc.) clearly showed how using multiple feature types can improve retrieval performance. Indeed, the optimal feature configuration for audio-only features yielded an optimal f-value of 44.3%, while a hybrid feature selection of 15 audio and 4 symbolic features scored an f-value of 61.1% [6]. This improvement shows the potential of using multimodal features, but it also shows that lyrical features did not help to improve system performance in this particular study.

Other studies, on the other hand, have highlighted the positive impact that lyrical features can make in MIR/MSA. In [10], Hu and Downie incorporate lyrical features into their testing

[1] Musical Instrument Digital Interface: digital music format designed for symbolic music representation and processing by computers.

[2] also known as

[3] Support Vector Machine

[4] K-Nearest Neighbor

and report a 9.6% accuracy improvement over the best audio-features-only system they tested. Few approaches in [9] [8] have suggested considering user profiles, moods, and context information, in addition to content-based and textual music features, to generate sentiment-aware and contextually meaningful music playlists. Therefore, we can see that the latest trend: i.e., performing sentiment analysis using multiple feature values; is receiving more interest and tends to produce better results. Yet, one can also realize, given the results just described and the relative novelty of MSA research, that this domain is still very much in flux as more progress is expected in the upcoming years.

IV. SYSTEM ARCHITECTURE

MUSE is designed and developed to allow users to predict the emotional response of a given musical piece. It leverages several cutting-edge algorithms and blends them with a music-theoretical knowledge base to infer the sentiment response from a composition's melody. The overall architecture of our prototype system is shown in Fig. 1. The *MUSE* engine consists of three main components:

1- Feature Parsing component: It receives an input MIDI file and returns a feature vector comprising of a combination of seven (4 *symbolic* and 3 *frequency-domain*) features, namely:

- **High-level symbolic features:**

1) *Note density (ND):* The number of notes per musical beat.
2) *Note onset density (NOD):* The number of distinct note onsets per musical beat. This feature differs from the previous one in that two notes played simultaneously count as one onset in computations. This feature indicates how the notes of a particular piece are played: If ND and NOD are similar, then we can infer that the notes in a piece tend to be played sequentially rather than together.
3) *Dominant key:* The key that is most common and most prominent in the musical piece. This feature is extracted using an approach similar to the one developed in [11].
4) *Chord progression:* The set of chords that best describe the musical melody. This feature is the most difficult to extract, and requires the use of a heuristic and a maximum-likelihood inference method to achieve satisfactory performance

- **Low-level frequency-domain features:**

5) *Piece Tempo:* The overall rhythm/speed of a musical piece (expressed in Beats per Minute (BPM)).
6) *Average pitch:* A weighted average of every MIDI note's pitch value, with the weight being the note's duration. This feature provides an indication of the overall pitch at which the musical piece's notes are being played in the frequency domain (designating high, medium, or low-pitch).
7) *Average intensity:* A weighted average of every MIDI note's velocity value, with the weight being the note's duration. This features indicates the overall intensity of a piece (e.g., calm or loud).

These seven features will be used to infer sentiments at the Machine Learning (ML) stage.

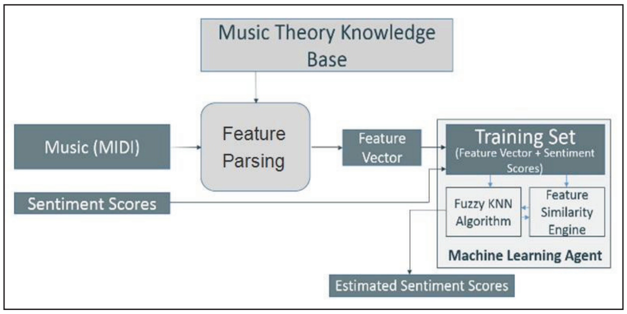
Music Theory Knowledge Base
Music (MIDI)
Feature Parsing
Feature Vector
Training Set
(Feature Vector + Sentiment Scores)
Sentiment Scores
Fuzzy KNN Algorithm
Feature Similarity Engine
Machine Learning Agent
Estimated Sentiment Scores

Fig. 1. Sentiment Engine Architecture

2- Music Theory Knowledge Base component: It houses all of the music theoretical operations, rules, and parameters needed throughout the system's operation. It is mainly called upon to perform likelihood estimations needed for MIDI feature parsing.

3- Machine Learning (ML) agent: this component is at the core of *MUSE*'s sentiment inference functionality. It consists of two sub-components and a training set:

1) *Fuzzy K-Nearest Neighbors (Fuzzy k-NN)* component: It implements a basic supervised learning algorithm allowing to compute sentiment scores for new incoming pieces based on their similarity with pieces it already learned. Unlike the traditional crisp k-NN algorithm (which classifies data in crisp/distinct categories), this algorithm produces fuzzy sentiment membership scores (producing so-called fuzzy categories with fuzzy boundaries, such that an object (musical piece) can be in one category and the other at the same time), which is more in keeping with the nature and subjectivity of sentiments (e.g., a piece of music can express 30% happiness and 70% excitement simultaneously).
While fuzzy logic can be integrated in other classifiers, we adopt fuzzy k-NN in this study since it is well established, non-parametric (usually allowing more flexibility and better performance, compared with parametric solutions), and instance-based (matching new pieces with existing ones using minimal training time).
2) *Music Similarity Evaluation* component**:** It allows the Fuzzy k-NN component to perform its estimations. At the most basic level, it accepts two input MIDI files and returns a similarity score ∈ [0, 1] highlighting their similarity or divergence (0/1 designating minimum/maximum similarity respectively). MIDI files are compared based on their symbolic and frequency-domain feature vectors. Individual feature vector similarity scores are computed using adapted similarity measures, such as Jaccard distance (used with most features) and the more sophisticated Tonal Pitch Step Distance (TPSD) used to compare chord progressions [12]. Then, the seven feature similarity scores are averaged to produce an overall similarity score.
3) *Training Set:* This forms the basis through which the ML agent can make estimations, providing the "expertise" the agent uses to infer incoming pieces' sentiment scores. The *MUSE* training set initially consists of 40 musical pieces which were annotated with the help of 30 human

testers via dedicated online surveys[1]. The initial 40-piece set is also diverse in that it covers all 6 primary emotions addressed in the *MUSE* approach: anger, fear, joy, love, sadness, and surprise. Beyond this initial training set, *MUSE* can be further trained on additional MIDI pieces using the system's lifelong learning feature. This will allow the tool to learn from bad estimations, as well as increase its own knowledge and overall user confidence in its ratings with system usage.

A diagram describing the ML agent is shown in Fig. 2.

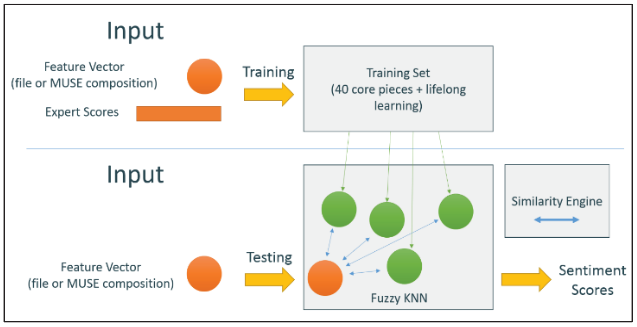

Fig. 2. *MUSE* Machine Learning Agent

V. EXPERIMENTAL EVALUATION

We have experimentally tested the different components of the system: from feature parsing, feature similarity evaluation, and primarily the ML agent, in order to assess their effectiveness and efficiency. We present in this paper results evaluating the ML agent only (and report the latter to a dedicated paper).

To perform the required testing, a suitable training set had to be constructed. Initially, twenty-four pieces formed the learning component's training set. These real pieces, ranging from classical to contemporary, were assembled into an online survey[1], where respondents were asked to rate each piece in terms of six sentiments (i.e., anger, fear, joy, love, sadness, and surprise) on a scale of 0-to-10. The survey produced around 30 responses, the average of which was used to train the system. At this stage, the learning component scores produced a PCC[2] of 0.53 using three-fold cross validation (i.e., 16 training pieces, 8 testing pieces). Seeing that the result was unsatisfactory, we proceeded to increase the size of the training set to 100 pieces by producing 76 "synthetic" pieces using a dedicated sentiment-based music composition tool[3]. These pieces were added to the system's training set using the lifelong learning feature. Using 10-fold cross validation, we obtained a PCC of 0.67, which is a remarkable improvement over the 0.53 figure mentioned previously.

However, after closely analyzing the data and results, we identified another issue with our training set: it seems *biased* toward certain emotions. Indeed, our set was overwhelmingly made of joyful and sad pieces, while angry, fearful and surprising pieces were almost nonexistent. To remedy this situation, we added an additional 16 real pieces to the training set, expressing mostly anger and fear. These pieces were selected based on human sentiment scores obtained by averaging the results of two other online surveys designed in a similar format to the first survey for the first 24 pieces[1]. We also dealt with inconsistencies in piece ratings by eliminating scores with the least inter-tester score correlations. Finally, we tackled the bias issue further by removing 16 sad and joyful pieces from the 76-piece synthetic set, whilst replacing them with 20 more evenly distributed pieces.

The resulting set, when looked at in a crisp manner (i.e., assigning a piece to the crisp sentiment category corresponding to the maximum sentiment score), had the following distribution:

Anger: 17, Fear: 17, Joy: 26, Love: 18, Sadness: 25, Surprise: 17.

For this final training set, we obtained a PCC of 0.63 using 10-fold cross validation, which proved to perform better on a wider range of musical pieces. Therefore, we converged on the 120-piece training set described above and proceeded to test our system using this set in terms of both efficiency and effectiveness.

A. *System Effectiveness*

To assess the quality of our system, we conducted tests covering our ML component's fuzzy scoring ability. We tested the ML agent using measures like the Pearson Correlation Coefficient (PCC) and Mean Square Error (MSE) compared with human tester scores. PCC is a correlation measure and evaluates the dependence between vector shapes ($\in[-1, 1]$, i.e., 1 for maximum correlation, 0 for no correlation, and -1 for negative correlation), whereas MSE is a distance measure evaluating the separation between vectors (as their average Euclidian distance $\in[0, \infty[$). A high quality sentiment analysis (classifier) would naturally produce: i) high PCC scores: which means that system generated sentiment vectors are closely correlated with user (expert) vectors; ii) and low MSE scores: meaning that system generated vectors are not distant from expert vectors. Experimental evaluation was conducted using 2, 3, 5, and 10-fold cross validation. The average results of these tests can be seen in Fig. 3 and Fig. 4

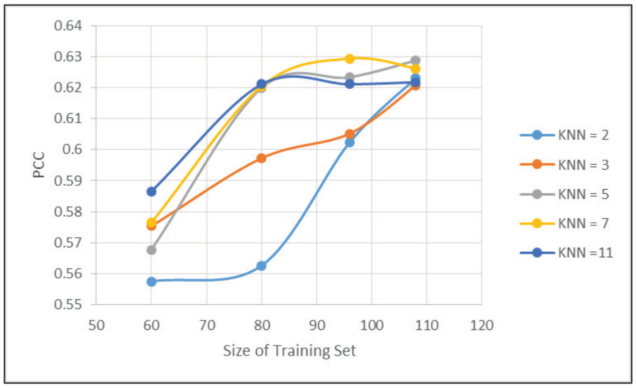

Fig. 3. PCC versus Training Set Size

From these results, we can make two observations. First, we clearly see that system performance improves as the size of the training set increases, and this for both PCC (steadily increasing) and MSE (steadily decreasing).This shows that *MUSE*'s ability to extract sentiments from musical pieces improves as it is exposed to and as it learns more and more pieces. Second, we also notice from the figures that PCC tends to increase as k-NN (the number of nearest neighbors considered in the Fuzzy k-NN classification process) increases, while MSE drops with the increase of k-NN. Following our analysis and understanding of Fuzzy k-NN, as we increase the number of neighbors, the training vectors used for score computation become more diverse and less similar to the target piece's vector (increasing the learner's training set variety, and thus increasing its resistance to noise when performing

[1] Available on https://goo.gl/forms/ptMy5uxrVQVmro5F3 (first part, 24-pieces); https://goo.gl/forms/tHFqeCvGBe7Nh2um1 (second part, 8-pieces); and https://goo.gl/forms/sOTjPJ986MGYjtsK2 (third part, 8-pieces).

[2] Pearson Correlation Coefficient

[3] Available at: http://sigappfr.acm.org/Projects/MUSEC/

classification). Hence, training vectors become more normally distributed, which in turn reduces and normalizes the system generated sentiment vectors.

B. *System Efficiency*

The Fuzzy k-NN algorithm requires no training time since it is non-parametric. In other words, training the system merely consists of adding an element to its training set, which is done in constant time. Though this speed in training is very advantageous, it comes at the expense of testing speed. Where other learning algorithms run in near instantaneous time following a lengthy training and parameter computation, the k-NN algorithm's running time is linear w.r.t. (with respect to) the size of its training set, since it must compare the target vector with each and every piece in its training set. Hence, what k-NN gives in training (in terms of efficiency), it takes back in testing.

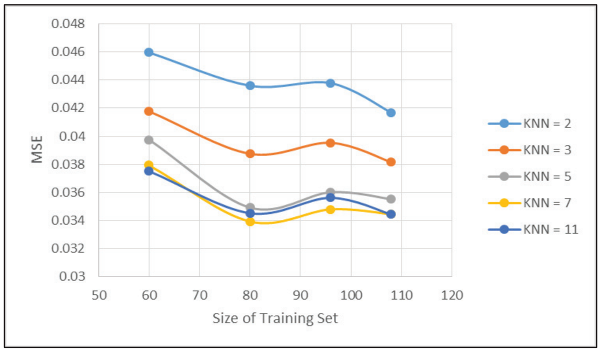

Fig. 4. MSE vs Training Set Size

Fig. 5. shows the system's time performance with different training set sizes, by varying *k* (i.e., the number of *k* nearest neighbors the system takes into consideration when computing scores). As expected, results show that the algorithm's running time is linear w.r.t. training set size as well as *k*, where increasing the *k* value leads to a larger overhead due to the added computations needed to consider the additional neighbors.

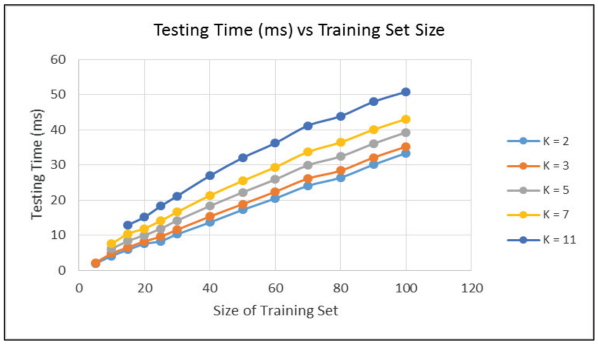

Fig. 5. System Running Time for different training set sizes and *k* (nearest neighbor) values.

VI. CONCLUSION

This paper briefly describes *MUSE*: a prototype system for automated MUsical Sentiment Expression. It mainly consists of: i) a feature parsing engine used to extract high-level (symbolic) music features from an input MIDI file, ii) a music-theoretic knowledge base to help with the music feature parsing process, as well as iii) an ML component tasked with converting music feature scores into sentiment vector scores. Developing this prototype system required conducting a thorough review of the literature in Music Information Retrieval (MIR) and in Music Sentiment Analysis (MSA). It was through this review that the features to be used in our approach were selected. Then, the system architecture was designed, and incrementally refined. With the system design in mind, we proceeded to implement the system and find the best starting set to set it up for as general an input file as possible. We then conducted a battery of experimental tests to evaluate the quality and performance of the system.

In the oral demonstration of *MUSE*, we aim to showcase the system's logical design, implementation, and functionality: defining and then fine-tuning the different system parameters, and then highlighting their impact with respect to the musical pieces being tested. We will also present and discuss our latest experimental evaluation and results, highlighting the system's effectiveness and efficiency, as well as its strong and weak points in extracting fuzzy sentiment scores and crisp sentiment categories, emphasizing ongoing design and technical improvements.

Looking forward, we plan to extend *MUSE* to consider a wider range of high-level (symbolic) and low-level (spectral) music features, in addition to the seven features currently used, aiming to further improve sentiment expression accuracy. In the near future, we aim to customize the system's behavior to consider a specific user's profile and her individual perception of sentiments in music. In other words, while our current system considers average user (expert) scores in training the machine learning agent, in order to produce scores that simulate the combined perception of all users (i.e., simulating the sentiment perception of the "mass" of users), we aim to extend/adapt the current work to simulate an "individual" user's perception of sentiments in music, based on her profile, preferences, and previous experiences.

ACKNOWLEDGEMENTS

This study is partly funded by the National Council for Scientific Research (CNRS-L) - Lebanon, and LAU. We would also like to thank the testers who volunteered to participate in the experimental evaluation.

REFERENCES

[1] Pennebacker Conglomerates, Inc., [Online]. Available: http://liwc.wpengine.com/. [Accessed Jan. 2018].

[2] IBM, [Online]. Available: https://www.ibm.com/watson/ developercloud/tone-analyzer.html. [Accessed Jan. 2018].

[3] MIDI Manufacturers Association, "The MIDI 1.0 Specification," [Online]. Available: https://www.midi.org/ specifications/category/midi-1-0-detailed-specifications. [Accessed Jan. 2018].

[4] Demopoulos R. and Katchabaw M. J., "Music Information Retrieval: A Survey of Issues and Approaches". *Technical Report #677, Dept. of Computer Science, Unviersity of Western Ontario*, Canada, Jan. 2007.

[5] Schedl M. *et al.*, "Music Information Retrieval: Recent Developments and Applications," *Foundations and Trends in IR,* 8(2-3), 127-161, 2014.

[6] Panda R. *et al.*, "Multi-Modal Music Emotion Recognition: A New Dataset, Methodology and Comparative Analysis". *10th Inter. Symp. on Computer Music Multidisciplinary Research (CMMR)*, 2013.

[7] Katayose H. *et al.*, "An Approach to an Artificial Music Expert". *Inter. Computer Music Conference (ICMC'89)*, pp. 139-146, 1989.

[8] Wohlfahrt-Laymanna J. and A. Heimburgerh, "Content Aware Music Analysis with Multi-Dimensional Similarity Measure". *Information Modelling and Knowledge Bases XXVIII,* pp. 292-303, 2017.

[9] Fleischman M. B. and Roy D. K., "Estimating Social Interest in time-based media". *US Patent 20110040760 A1*, 2011.

[10] Hu X., "Improving Mood Classification in Music Digital Libraries". In *10th annual joint conf. on Digital Libraries, ACM*, 2010.

[11] Temperley D., "A Bayesian Key-Finding Model". *MIREX-2005 Symbolic Key Finding Task*, 2005.

[12] de Haas W. B. *et al.*, "Tonal Pitch Step Distance: A Similarity Measure for Chord Progressions". *Intern. Conf. on Music IR (ISMIR)*, 51-56, 2008.