

LyQ - An Emotion-aware Music Player

David Chawei Hsu
iSpace Laboratories
National Taiwan University
chawei.hsu@gmail.com

Jane Yung-jen Hsu
iSpace Laboratories
National Taiwan University
yjhsu@csie.ntu.edu.tw

Abstract

This paper introduces LyQ¹, a sensible music player that may change the way you experience music. Music listening will no longer be passive like the radio, nor manipulative like hand-picked tunes on your stereo. LyQ provides music-on-demand based on a single line of input in natural language to describe what you want for the moment. For example, enter "I am throwing a dance party tonight!", and LyQ will recommend the best selections from your music library suitable for playing at your party. To sense the emotion of a musical piece, LyQ analyzes its melody, lyrics, and metadata. In particular, the current implementation conducts textual analysis with commonsense technology. This paper presents the overall design of LyQ as well as a brief overview of various emotion representations, commonsense technology, and development tools.

Introduction

Imagine an opportunity for your claim to fame. You have volunteered to be the DJ for an upcoming dance party on campus, and your first task is to put together a playlist. How would you do it? You may use "party" as a keyword to search in your personal collection or online. Alternatively, you may go through your music library and sift out what you like. Both approaches are neither efficient nor effective. It would be ideal if one could simply enter a description in natural language, such as "I'd like some happy songs", "I'm feeling blue.", or "I am throwing a dance party tonight!", and a sensible choice will be generated automatically. This paper introduces LyQ, a commonsense music player, which recommends songs that reflect the desired emotion as expressed in the user input, either explicitly or implied.

Many people have amassed a large collection of music, but may not always have the time to enjoy pieces not on their "top rated" or "frequently-played" lists. To explore unfamiliar yet interesting songs from their own music collection or shared music from friends, it is useful to enable an intuitive way of finding music based on a user's profile, ratings, music-playing history and interests.

Copyright © 2006, American Association for Artificial Intelligence (www.aaai.org). All rights reserved.

¹This research is sponsored in part by a grant from Intel and the National Science Council of Taiwan (NSC94-2218-E-002-057).

LyQ is designed to explore affect sensing for music based on existing psychological, semantic, and commonsense theories, as well as our personal experience in music playing. In general, a song consists of *melody* and *lyrics*. Melody refers to the audio content comprising both instrumental and vocal tones, while lyrics refer to the textual content of the song. Both melody and lyrics convey the emotional substrate of the musical creation. Each song is also associated with its *metadata*, which includes title, artist, album, date and reviews. By conducting textual analysis with commonsense on lyrics and metadata, LyQ is able to compute the emotion model of a given song.

This paper gives a brief overview of emotion representations, commonsense technology, and music analysis tools. The design, architecture, emotion model in music, and user interface of LyQ will be presented in the following sections.

Background

Before we introduce the overall system architecture, let us examine two key concepts in LyQ. First, we'd like to capture the *emotion* of a given song, in addition to its artists, title and other basic information. Second, we believe the emotional context of a song can be revealed via textual analysis of lyrics by using commonsense technology.

Music and Emotion

A suitable model for representing emotions is essential for the proposed emotion-based recommendation system. In the following, we survey three different affect representations and discuss their pros and cons.

Paul Ekman's six universal facial emotions This affect ontology describes the six emotions: *happy*, *sad*, *angry*, *scared*, *disgusted*, and *surprised*. This representation is adopted in the `guess_mood` function of ConceptNet (Liu & Singh 2004). While these six attributes can be used in varying combinations and intensities, it is somewhat not suitable for classifying music. We need a richer representation to express a wide range of emotions conveyed in music.

Albert Mehrabian's PAD The PAD representation defines the three dimensions of affect to be *pleasure-displeasure*, *arousal-nonarousal* and *dominance-submissiveness* (Mehrabian 1995). PAD is extremely

versatile and can be used to express a wide range of emotions. For example, Ekman’s six emotions can be easily described in terms of PAD values. When using this representation in a musical domain, it’s better to ignore the third dimension of dominance and focus on the relationship between pleasure and arousal, since we can see this observation in Affective Listener (Chung & Vercoe 2005) where positive pleasure and positive arousal results in an engaging affective state. In contrast, a negative value in both pleasure and arousal dimensions locates in a boring affective state. The drawback of this representation is that all possible moods need to be carefully mapped onto this two-dimensional emotional coordinate.

Kate Hevner’s Adjective Circle The Kate Hevner’s music-emotion representation maps many musical parameters onto a circle of emotional terms (Hevner 1936). The circle as shown in Figure 1 is divided into eight groups of closely related emotional adjectives. The eight emotional states are angry, exciting, happy, playful, relaxed, sad, spiritual, and tender. This representation is better suited to musical applications than the previous two because it provides a number of musical parameters to describe each mood.



Figure 1: Kate Hevner’s Adjective Circle

Commonsense Technology

Lyrics can be viewed as a concrete, implicit expression of an abstract concept, which often makes listeners catch the feeling of a song even without listening to it actually. Given that lyrics are written in natural language by people, its content can be analyzed with the help of some commonsense knowledge base. Commonsense knowledge, thusly defined, spans a huge portion of human experience, encompassing knowledge about the spatial, physical, social, temporal, and psychological aspects of typical everyday life. As a result, it makes sense to take advantage of ConceptNet’s affect-sensing capabilities in creating this application.

Open Mind Common Sense (OMCS) is currently the second largest commonsense database in the world (Singh et

al. 2002). This generic commonsense knowledge base is contributed by web volunteers entering their commonsense statements into the OMCS corpus. Until now, they have entered over 700,000 such statements. With this knowledge it is possible to create innovative software that can solve everyday situations and problems or even produce a brand new interaction with computers by using commonsense.

ConceptNet (Liu & Singh 2004) is a commonsense knowledge base and natural-language-processing toolkit and it automatically generates a commonsense network by using the commonsense statements in the OMCS database. This program offers many useful functions, such as topic-gisting, affect-sensing, analogy-making, contextual expansion, etc..

Implementation

This section introduces the detailed design of our music player as shown in Figure 2. The system is written in Python, a high-level object-oriented language, and operated on the Mac OS X platform. Following Figure 2, there are three different types of input - query, textual content, and acoustic content. A query is described in natural language and given by the user. For example, a user may input a sentence such as “I’m throwing a dance party tonight!” that shows the user’s emotion and desire. For any given song, its *textual content* includes all textual components such as metadata and lyrics; its *acoustic content* refers to the audio component of the song. To extract important features, the LyQ Analyzer adopts different models of evaluation based on the input type. These extracted features are stored in the Database, implemented in MySQL. According to the user’s query, the Matcher searches for the top-five songs with similar implicit feeling and semantic meaning as the query sentence. Finally, LyQ recommends the matching songs to the user.

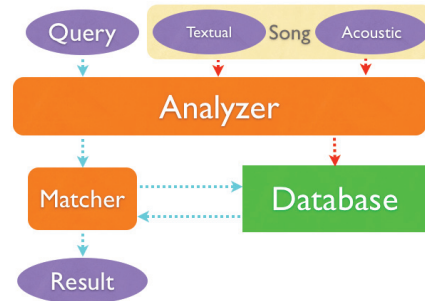


Figure 2: System Architecture

Textual Affect-sensing Model in Music Domain

This is the core part of the Analyzer with respect to textual analysis. After surveying several emotional representations, we decided to use Kate Hevner’s Adjective Circle to construct our affect-sensing model. An eight-tuple is used to represent the dimensions of music mood, including *angry, exciting, happy, playful, relaxed, sad, spiritual, and tender*.

Regardless of whether the input is a document or a sentence, the end result will be represented as an emotion vector:

[*a* angry, *b* exciting, *c* happy, *d* playful, *e* relaxed, *f* sad, *g* spiritual, *h* tender]

Each of these eight entries ranges from 0 to 1, representing the magnitude of each particular emotion.

Generating node-score file Following the affect sensing approach by Liu et al (Liu, Lieberman, Selker 2003), we propose to achieve music mood inferencing by carrying out spreading activation in ConceptNet. For each iteration during spreading activation, the music mood value of a node in ConceptNet is propagated outwards to its neighboring nodes with a discount factor $d = 0.25$ in the prototype system.

A total of 30 descriptive words are defined as the *emotion grounds*. For example, we define the word “party” to be one emotion ground with a tuple value [0.3, 0.8, 0.7, 0.6, 0.3, 0.1, 0.0, 0.5]. The emotion grounds begin propagating their values in ConceptNet using spreading activation. In previous experiments, we tested the `guess_mood` function in ConceptNet with `max_node_visits= 1000`, which is a parameter determining how far the context will spread. The response time may take 10 seconds approximately. To achieve our expected performance, the scores computed by spreading activation are cached in the node-score file.

Get the mood Any sentence is an input no matter query, song title, or a sentence in lyrics. We obtain an extraction object which contains a parsed digest of the text from the `get_extraction` function. Afterward, by matching the same term which occurs in the node-score file, we receive this emotion value. The flow diagram of getting mood features is clearly illustrated in Figure 3.

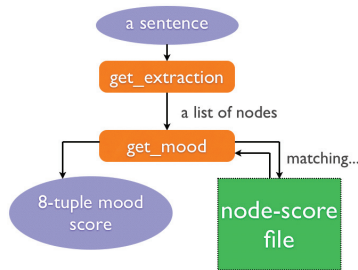


Figure 3: Flow Diagram of Getting Mood Features

Since lyrics can be considered as a special document or maybe a poem which has been merged with a number of sentences which show varying strength of emotions in every paragraph, we memorize each sentence’s mood and position in order to generate a global model with better prediction.

Generating Dataset

In this section, we introduce our database and dataset generation. The setup is simple and intuitive. The first thing you need to do is to set the location path to your music folder to avoid having the system scanning over the entire storage.

A local database is created automatically. Before rebuilding your local database, the system will check out any new audio files that have never been analyzed previously. Otherwise, the local database will connect to the Internet and load the default database from the LyQ website. This design produces high scalability, because all the necessary steps before running the application are automatic.

After the system confirms the checklist that needs to be analyzed, if the computer is online, the system will search and obtain the metadata and lyrics from several available websites which provide the information. Subsequently, by sending this information to the Analyzer, LyQ generates and returns extracted features to the Database.

Currently, we have collected 225 songs in our dataset, including famous bands like *the Beatles*, *Jack Johnson*, *No Doubt*, *Red Hot Chili Peppers*, *Blur*, and many other different styles of performers. However, this dataset is still limited in several genres, thus our next step is trying to collect more various genres to test our system.

User Interface

Our interface is implemented by using Macromedia Flash and Actionscript language which have the advantage in developing a user-friendly interface. This interface connects with our main system via XML-RPC. In our first prototype shown in Figure 4, users can simply enter one sentence in natural language into the text entry under “Query” label, and then press the “Search” button. Immediately, the system recommends at least 5 songs at the right-hand music player that have the feeling similar to your sentence implied, and also displays the mood values of your input sentence on the orange “Result” text field.

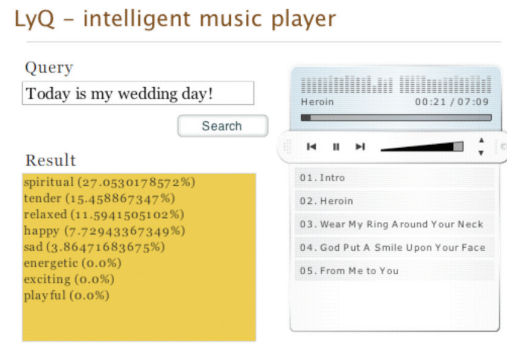


Figure 4: LyQ’s User Interface

User Study

Music comes from human beings, thus we desire music. In order to get the feedback, we invited several different types of music lovers to use our system. Their first reactions were similar, most users thought it is an interesting and amazing music player, especially the way of searching. But some of

them were still not familiar with this searching style. After our discussion, we bring out several ideas. First, based on what behaviors users already have in traditional music search, they remain using their favorite performers or songs to describe what kind of music they love. Second, people listen to music with desires. For example, when a person is busy, he'd like to listen to the songs that makes him relaxed.

In summary, on the next music-on-demand generation, we shall improve our interface and recommendation strategy to adapt users' manners. For example, provide some frequently existing options like genres, popularity or optionally ask users "How do you feel right now?", "Why do you want some music?", "Where do you gonna play?", "Whom are you playing to?", etc.. Those interactive ways might be more intuitive and practical.

However, since we cannot suddenly change one's mind, we offer an innovative interface and concept for users and recommend more suitable songs to them or even additionally allure them to listen to variety of music according to their backgrounds and profiles.

Related Work

Emotion-based music classification is gradually coming into view on the web, such as All Music Guide, it provides more than 210 emotional categories for classifying entire songs. Each song is not only classified into one category, but can be tagged with multiple categories, that means those categories are not exclusive and also mentions that a realistic classification system should relax the strict classification paradigm. It provides a brand new way of classifying and searching music, but does not solve the problem of finding adequate music for users.

Another music channel is listening to *Internet Radio*, such as Yahoo! Music, Pandora, etc.. By entering an artist name or song title, those online systems will create a personal radio station for you and play songs which have similar features and characteristics to the artist or the song title you gave. According to the Pandora website, they claim that they capture the unique and magical musical identity of a song from melody, harmony and rhythm, to instrumentation, orchestration, arrangement, lyrics, and the rich world of singing and vocal harmony. It seems like a really creative project but requires lots of human resources in development.

Finally, a commonsense playlist generator, MySoundTrack (Meyers 2005), created by Owen Meyers in MIT Media Lab is carrying the similar idea as mine in creating an innovative music recommendation system. Currently, the functions we provide are in the homogeneous level, except the self-organizing ability and ease-to-setup attribute in LyQ. In our following experiment, it is essential to compare the results and the performance with those similar systems in order to obtain a better solution.

Conclusion and Future Work

In this paper, we introduced an innovative interface to access your personal music collection. Instead of the traditional keyword-based search, LyQ offers a novel emotion-based search interface that attempts to capture a deeper feeling in

a song. The affect-sensing approach supports users to indirectly observe the relationship between melody and lyrics. As a result, LyQ improves music emotion detection, which is usually based on acoustic analysis in Music Information Retrieval.

In the current stage, we have accomplished the textual analysis components of LyQ, and our next target is to construct the acoustic analysis component. To assure the accuracy of LyQ, we need to combine both textual and acoustic models. On the other hand, people have different behaviors in music listening. Some prefer to pay more attention in melody but some in lyrics; some prefer popular but limited range of songs, while others prefer alternatives and a wide range of songs. Therefore, we should consider giving different weights to achieve personality, according to their profiles and preferences. We have mentioned various listening experiences that may be considered in music entertainment of the future. Our goal is to break up users' present music-listening behaviors and rules in exploring their own musical interests.

References

- Chung, J., and Vercoe, S. 2005. The affective listener. <http://courses.media.mit.edu/2005spring/mas630/05.projects/AffectListener/index.html>.
- Hevner, K. 1936. Experimental studies of the elements of expression in music. In *American Journal of Psychology*, 48 (1936), 246V268.
- Liu, H., and Singh, P. 2004. Conceptnet: A practical commonsense reasoning toolkit. In *BT Technology Journal, To Appear. Volume 22, forthcoming issue. Kluwer Academic Publishers*.
- Liu, H.; Lieberman, H.; and Selker, T. 2003. A model of textual affect sensing using real-world knowledge. In *In Proceedings of the Seventh International Conference on Intelligent User Interfaces, pages 125-132, 2003*.
- Mehrabian, A. 1995. The pad comprehensive emotion (affect, feeling) tests. <http://www.kaaj.com/psych/scales/emotion.html>.
- Meyers, O. 2005. Mysoundtrack: A commonsense playlist generator. <http://web.media.mit.edu/~meyers/mysoundtrack.pdf>.
- Singh, P.; Lin, T.; Mueller, E.; Lim, G.; Perkins, T.; and Zhu, W. 2002. Open mind common sense: Knowledge acquisition from the general public. In *Proceedings of the First International Conference on Ontologies, Databases, and Applications of Semantics for Large Scale Information Systems. Lecture Notes in Computer Science (Volume 2519). Heidelberg: Springer-Verlag*.
- University, T. L. 2003. Detecting emotion in music. <http://citeseer.ist.psu.edu/645004.html>.
- van de Laar, B. 2006. Emotion detection in music, a survey.
- Yang, D., and Lee, W. 2004. Disambiguating music emotion using software agents. In *ISMIR*. <http://ismir2004.ismir.net/proceedings/p011-page-52-paper218.pdf>.