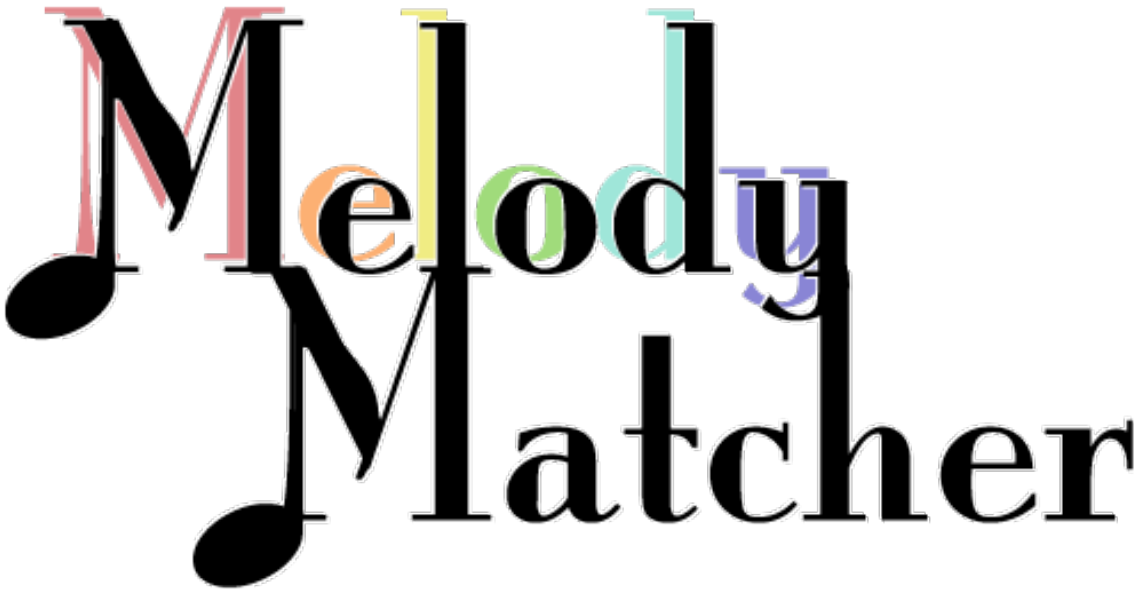

Jenee Hughes

CSC 590

Dr. Gene Fisher

Fall 2010

Keywords, ACM Taxonomy: **10,V,VII** Music; **8,V,V** Sound and Music Computing, **II** Modelling;**, II, 1, VI** Games and infotainment; **II,1,VII,IV** Language parsing and understanding; **II,1,VII,VIII** Text analysis; **II,1,VII,III** Language models;

Brief Summary:

Melody Matcher is a semi-automated music composition support program. It analyzes English lyrics along with a melody, and alerts the composer of the locations in the song where the lyrics are not understandable.

Basically, it's grammar- and spell-check for songs. This is significant, because no research had been done specifically on the quantifiable measurement of English-language lyric understandability, prior to my project.

Target Audience and Goals:

This program will be used as a compositional aid by anyone who wants to write songs and make them sound good, technically. Its should allow the song writer to focus on more subjective criteria of what makes a song "good", because it will make the structural rules of lyric composition immediately apparent.

My hope for this project is that it will be useful to burgeoning songwriters, who have the creative spark to make wonderfully poetic lyrics, but lack the "ear" to match their lyrics successfully to music. It should be particularly helpful to songwriters who place a high emphasis on understandability of lyrics (such as parody song writers, or lyricists for musical theater).

Additionally, Melody Matcher will be useful for songwriters for whom English is a second language. They may be a master lyricist in their native language, but writing lyrics in English can be a particular challenge, since so much of lyric-writing is dependent upon knowing the cadence of the language you're writing lyrics in, and since English has no easily-discernible rules for emphasis placement in words.

Basic Idea:

Melody Matcher aims to replicate the human ability to identify lyrics in a song that are easily misheard. Lyrics can be misunderstood due to:

- Lyric/Music emphasis mismatch, due to:
    - Note intervals
    - Phrase emphases
    - Word emphases
- Word "cramming", due to
    - syllable lengths that exceed that of note length
    - Mouth movement delta time intervals
- Word misidentification, due to
    - Altered pronunciation of words
    - Phone similarity
        - Voicing (voiced vs. voiceless)
        - Beginning/end mouth positions
        - Type (Plosive, Fricative, affricate, nasal, lateral, approximant, semivowel)
    - Improper cognitive parsing of phone sequences
    - Phone sequences that can be parsed more than one way
    - Non-deterministic interpretations of phone sequences.
    - Phone sequences with multiple interpretations and/or tokenizations

I won't go into detail on these, as that is outside of the scope of this paper, but suffice to say that I did a lot of work on fully and completely specifying these concepts, among others.

Related Work:

There are two types of work related to this subject. The first, and most profilic, refers to the subjective, or non-algorithmic, measurement of bad lyrical fit. The second, which is much rarer, refers to an algorithmic description of lyrical fit. I'll briefly explore both types.

Foundational/Theoretical / Subjective

Lyricists Books

Most of the current literature done on understandability of lyrics is written by Broadway lyricists with no particular technical background. Thus, it focuses less on measurably quantifying the understandability of a lyric, and more on subjectively identifying lyrics that "sound bad". Though most of this work doesn't speak in terms of overarching algorithms, reading through it provided me with expert input on songwriting. The most useful book I found was [1] G. Lees, *The Modern Rhyming Dictionary*, Cherry Lane Music, 1987. In it, I found the foundations of the idea that some consonants are more difficult to sing than others, among other things.
Other books in which I found useful, if unexplored, concepts were "Lyrics on Several Occasions", by Ira Gershwin; and "Lyrics", by Oscar Hammerstein, both of Broadway fame.

Analytical/Computational / Objective

When I started this project, no work had been published regarding means to algorithmically measure lyrical fit with English lyrics. There were a few similar systems, however.

Tra-la-lyrics[i]

The Tra-la-lyrics[ii] paper describes the most similar system to mine in existence prior to the start of my work. It takes lyrics in Portuguese, and matches them via emphases to the music. This is different from Melody Matcher, because Melody Matcher uses English-language lyrics, which is a significantly more complicated task, due to English's ambiguous links between phonetics and written symbols, and non-standard phrasing.

Multilingual Lyric Modeling and Management[iii]

This seemingly-unrelated-to-the-topic-at-hand article actually first introduced me to a usable paradigm for associating the music and lyrics in a song. Its main goal was to be able to substitute out lyrics in different languages, but the foundation it used to do so was extensible, and largely adopted by my program

Microsoft Research paper, "Relationships between lyrics and melody in popular music"[iv]

Much to my chagrin, upon investigation of prior work for this class, I discovered that Microsoft Research had released a paper exploring a concept similar to Melody Matcher, entitled "Relationships between lyrics and melody in popular music." Fortunately for me, their approach is woefully incomplete in comparison to mine, as it only deals with the effects of musical emphases on lyric understandability. The problem is significantly more complex than they made it out to be in the paper, and I hope to publish Melody Matcher as a research paper to share my findings.

Theoretical Validation Work So Far

Before diving headfirst into implementation, I needed to confirm that my algorithms were sound, theoretically. I did this by careful examination of a wealth of songs:

"The Bad Touch" (The Bloodhound Gang)

This song's fast-paced lyrics were nearly completely understandable to me on the first listen, making this a song that my algorithm should similarly find few faults with.
I used this song as my primary base case during the first quarter of development of Melody Matcher, developing much of my core algorithm via analysis of its lyrical and musical patterns. It was instrumental in my discovery of the importance of phones in my algorithm, allowing me to deeply investigate the ideas of **word length**, **lyrical cramming**, **lyrical bleeding**, and **deterministic cognitive parsing of lyrical phrases**, among other concepts.
The problem deriving my initial algorithm using this song, which I didn't identify until I was embarrassingly far into the process, is that this "song" is actually a rap. It has no note intervals. Much of my original analysis fell apart upon trying to apply my algorithm to songs with variable note intervals.

"One-Note Samba"(Jazz Standard)

Once identifying the note-interval-intolerance my initial algorithm, I needed to find songs with no note intervals, to validate my approach. This song, conveniently, has one note, used for all the lyrics in its verses.

"Tubthumping"(Chumbawumba)

This song was also used in the initial algorithmic extraction/determination/analysis/trying-to-figure-out-measurable-rules. It was instrumental in my exploration of the idea of **syllables having length and duration**. I discovered, during my second distillation/figureouting of the algorithm, that I had been using incorrect sheet music for the song. The version I was using had ignored the idea of flats and sharps, effectively eliminating most of the note intervals in the song. This

exacerbated the original algorithm's intolerance of note intervals, but has since been accounted for.

"Just Dance" and "Poker Face"(Lady Gaga)

These songs were integral in my exploration/discovery of the ideas of **phone similarity**, **non-deterministic mental parsing of phonetic sequences**, **word cramming**, **word bleeding**, and **word stretching**.

The word "hermitages", and "Strawberry Fields Forever" (The Beatles)

The selection of this song was prompted by my discovery of the word "hermitages", which has a *bizarre* pronunciation. Most words in the English language only have boolean emphases, where a syllable is either emphasized, or it's not. "Hermitages" is one of the few words in the English language with a **secondary emphasis**, which breaks the idea of boolean emphases. My algorithm at the time used boolean emphases, and gave incorrect results when encountering words that didn't fit with that paradigm. The word "strawberry" also happens to have a secondary emphasis, though the relative-uniqueness of its pronunciation doesn't strike us as odd, because its oddity is tempered by the frequency with which one encounters it in conversation. Though "hermitages" and "strawberry" both have non-standard pronunciations, "strawberry" is not perceived as weird-sounding.

This observation led to my decision to include the effects of **word-frequency/familarity** in my algorithm. There are a few different types of word frequency/familiarity: **pervasive frequency/familiarity** (in which a phonetic sequence is more like to be mentally parsed as word that is more common in everyday speech than one that is less-uttered), **title-based frequency/familiarity** (in which a phonetic sequence is more like to be mentally parsed as word that is in the title is. Example: the word "suburban" in the song "Penny Lane" by the Beatles), and **in-song frequency/familiarity** (in which a relatively-uncommon or even made-up word is correctly parsed by people's brains because of its disproportionate frequency in the song. Example: ***Copacabana***) When I get the time, I would like to write a parser that goes through a song's lyrics,

counts each word's frequency, and then compares it to the accepted universal frequency values.

"Penny Lane" (The Beatles)

The word "suburban" in the chorus is a good example of **in-song frequency/familiarity**.

"Blue" (Eiffel 65)

This should be a good example of **title-based frequency/familiarity**.

"Skullcrusher Mountain" (Jonathan Coulton)

The word "henchmen" in the chorus is a good example of **in-song frequency/familiarity**.

"Kokomo" (The Beach Boys)

All of the island names in the chorus make a good example of **in-song frequency/familiarity**.

"Like a Surgeon" and "Amish Paradise" (Weird Al Yankovic)

Good example of **title-based frequency/familiarity**. In Weird Al's song, "Like a Surgeon" the word "surgeon", which might normally be interpreted at the word "surging/surgin' ", is correctly interpreted, partially because of its presence in the title.

"Running with the Devil" (Van Halen)

"Sgt. Pepper's Lonely Hearts Club Band" (The Beatles)

"While My Guitar Gently Weeps" (The Beatles)

Implementation Validation Plans

Phase Alpha, Baby Steps : The Boolean Jenee Test

For Phase Alpha, Baby Steps, I will run a lyric and melody set through the parser, and see if it correctly identifies places in which I believe songs could be easily misunderstood.

Phase Alpha, Monkeys on Typewriters : The Boolean Random People Test

See if it correctly identifies where songs are misunderstood.
For Phase Alpha, Monkeys on Typewriters, I will play a song for a group of people, and have them write down what they think the lyrics are. Then, I will compare the incidence of misunderstanding to my measured coefficients.

Phase Alpha, Bender : The Boolean Experts Test

For Phase Alpha, Bender, I'll have a group of experts listen to annotate songs lyrics they believe to be difficult to understand. Then, I will compare the incidence of misunderstanding to my measured coefficients.

Phase Beta, Dogfooding: The Jenee Usability Testing

For Phase Beta, Dogfooding, I'll work with the GUI to see if I can understand what would be wrong with a song, and see how I can fix it.

Phase Beta, The Butterfly Room: The Random People Usability Testing

For Phase Beta, The Butterfly Room, I'll release the program to a select group, and monitor if they can understand what's happening with the program. Also, see if they can learn lyric-writing from it.

Phase Beta, The Caterpillar Room: The Experts Usability Testing

For Phase Beta, The Caterpillar Room, I'll release my program to expert lyricists, to let them tear it apart.

Phase Gamma, Profit: Jenee Starts A Business

For Phase Gamma, Profit, I'll build a business model, and see if my idea actually has merit in the real world, as measured by commercial success.

Glossary

Note Intervals: A note interval is a change in note pitch. If a song has no note intervals, it is probably a rap (or a song with one note)

Phones: A phone is an atomic sound in the English language. All vowels and consonants are made of one or more phones.

BIBLIOGRAPHY:

[i] H.R. Oliveira, F.A. Cardoso, and F.C. Pereira, "Tra-la-Lyrics: An approach to generate text based on rhythm," *Proceedings of the 4th. International Joint Workshop on Computational Creativity, London, UK*, 2007.

BIBLIOGRAPHY:

[ii] H.R. Oliveira, F.A. Cardoso, and F.C. Pereira, "Tra-la-Lyrics: An approach to generate text based on rhythm," *Proceedings of the 4th. International Joint Workshop on Computational Creativity, London, UK*, 2007.

[iii] P. Bellini, I. Bruno, and P. Nesi, *Multilingual Lyric Modeling and Management*, Visual Perception of Music Notation: On-Line and Off-Line Recognition, edited by Dr. Susan E. George, press, 2003.

[iv] E. Nichols, D. Morris, S. Basu, and C. Raphael, "Relationships between lyrics and melody in popular music," in Proceedings of the 10'th International Conference on Music Information Retrieval (ISMIR), 2009.