Lyric Generation using Artificial Intelligence

R.Sam Thayakaran#1, StephenJebakumar*2,

Department of Computer Applications, S.A.Engineering College,
rsblueheart@gmail.com, stephenchriz@gmail.com

Abstract—Unlike a poem that comes from the soul of the poet, lyrics are written to suit the situations in the stories and are sometimes set to the tunes which the Music-Director provides. There are also unique demands from producers as they want some catchy meaningless words of any language. Some software tools that will help me to realize our goal of lyric automation and computer aided lyric have already been published by various companies like “paadal” and “Emoni” , But the difficulty here is to create appropriate similes and metaphors. Lyric Generator, It’s an application that can be used to generate lyrics automatically i.e. all by itself. Here the user only needs to describe the scenario of the lyric that is the mood , relationship , character description , rhyming , similarities , inspirations for the lyrics. These are certain criteria by which the computer or the application understands about what exactly we want it to write for us. The lyric generator is an interesting topic for artificial intelligence; however it is very essential that we provide lot of knowledge to the system prior to the creation of the application. And it’s also important that we always need to constantly update the data base of the words as there are new words created regularly and the language pattern has been evolving continuously.

Keywords— *lyric, songs, pattern, rhyme, genre*

1. Introduction

The Automatic generation of lyric involves several levels of language like phonetics, lexical choice, syntax and semantics and usually demands a considerable amount of input of knowledge. It is a complex task and though it’s an interesting for artificial intelligence.

We start by presenting two different categories for the existing poetry generation system. The first is based on the approaches and techniques used to accomplish lyric generation and the second is based on the goals of the system and on output text. A brief overview on several systems capable of generating lyrics is provided.

2. Background

Lyric Creation by itself is a huge area in Computer science and Artificial Intelligence. We restrict our literature to music and Lyric Creation only. In[1], authors propose to visualize text readability using a few readability metrics. The technique provides means for readers to see which sections of text are hard to read with complicated words. Paadal [2], an interesting project, presents a method to automatically select icons for different sections of tunes. Apart from considering the musical features, Paadal considers text features to select icon based on the concept. ‘See the music’ [3] is a meaningful project which tries to visualize music features and emotions for the people hearing to the music. Authors of [4] present a visualization of lyrics. Here they use lyric features to extract an event and compose an image from the extracted concepts. The work in [5] presents a visualization technique with lyrics using musical features such as pitch to assist and instruct karaoke singers. In this paper, we propose visualization of a lyric purely based on ten features of lyrics. The ten features are mapped to features of a flower. The following section explains the ten features in detail.

3. Categories of Lyric Generating Systems

Despite not requiring exaggerate precision [1], poetic texts involve regular syntactic and phonetic patterns where rhythm, metrics, rhyme and other features like alliteration or figurative language play an important role. MadhanKaarky [5] affirms that when it comes to writing poetic text, we need to break several rules that are usually present in the production of natural language text. He refers some specific issues that need to be taken into consideration while writing poetry, High occurrence of interdependent linguistic phenomena that requires consideration of semantics, syntax and lexis, there may not be a well-defined message, and rich resources are needed to satisfy phonetics, syntaxes semantics. Objective evaluation of the output text is very difficult. This section presents two different proposed categorizations for poetry generation systems according first to the approach aken and techniques used and then to the goals the systems try to achieve. For each of the presented categories, examples of systems that embody its properties are given.

4. Lyric Generator

The Lyric Generator is a simple system that generates Lyrics based on words describing a subject, a synonym for the subject and a title for the Lyric, all three given by the user. The resulting text consists of pre-defined verse templates where gaps are filled with the words given. The lyric generates the lyric on the basis of the requirement specified by the user.

5. Approaches and Techniques

- Template Based Lyric Generator: Templates of lyric forms are filled with words that suit the defined constrains (either syntactical or rhyming or both).
- Evolutionary Method: Here we use the existing lyrics and follow its pattern to create a new lyric.
- Rhyming patterns: We can create lyrics through following a certain rhyming patterns.

6. Goals

For the purpose of his thesis, Manurung defines that poetic text must hold all the following three properties:

- Meaningfulness: convey a conceptual message, which is meaningful under some interpretation.
- Grammaticality: obey linguistic conventions prescribed by a given grammar and lexicon.
- Poeticness: exhibit poetic features.

7. Lyric Features

Independent of the language, a lyric is rich in features not found in prose. Lyric is usually a small piece of work compared to blogs or news articles. The following are the ten features that we identify lyrics and propose to use for the visualization in the next section. All statistical details mentioned in this section were obtained from a collected set of 1000 Tamil lyrics, which were created over a period of 70 years.

7.1.Song Length

Song lyrics do not vary much in length. Average length of a Tamil lyric is 38.56 lines, which we consider as the medium length. In addition, 4 windows of 5 lines on both sides classify songs as very short, short, long and very long.

7.2.Base Genre

Songs are usually classified into genres based on music. In this case, the base genre classifies the songs into 10 classes based on lyrical theme. Character Description, Romance, Philosophy, Festival, Occasion, Relationship, Nature, Patriotic, Spiritual or Miscellaneous are the 10 base genres

7.3.Mood Genre

Irrespective of the base genre, a lyric can have one of the six moods from Happy, Excited, Tender, Scared, Angry or Sad. The mood genre classifies the lyric based on the words pertaining to the different emotions.

7.4.Style Genre

Tamil has various dialects. The style genre classifies a lyric based on the language style. Traditional, Folk, Contemporary or Mixed are the classes in the style genre.

7.5.Simile Count

Tamil has various dialects. The style genre classifies a lyric based on the language style. Traditional, Folk, Contemporary or Mixed are the classes in the style genre.

7.6.Freshness

A lyric’s freshness can be computed in ways such as usage of words, similes, rhymes and many more ways. We consider just the usage of words over a period of ten-year time windows to compute the freshness of lyrics.

7.7.Pleasantness

Ignoring the music and meaning of lyrics, the sounds used in the words would determine the pleasantness of lyrics. Usage of nasals and long vowels increases the pleasantness of a word and thereby a lyric. Pleasantness is computed by modeling the place of articulation of phonemes and the consonant family variations in Tamil.

7.8.Unrecognized Words

Words that do not exist in a dictionary and cannot be analyzed by a morphological analyser are classified as unrecognizable words. We use atchayam [ref] morphological analyser to analyse Tamil words. Non-Tamil words also fall under this category.

7.9.Rhyming Strength

Rhyme is a key feature of lyrics. Lyrics which are rich in rhymes are popular among kids. Tamil has a very unique rhyme schemes and rhyme patterns in lyrics. First letter match (monai), Second letter match (edhugai) and End letter match (Iyaibu) are considered to compute a rhyme score in a lyric.

7.10.Dominant Rhyme

In some lyrics, edhugai or monai may be dominant. Most lyrics have iyaibu as their dominant rhyme. The dominant rhyme is identified by simple count of the rhyme schemes.

8. Requirement Analysis

The requirement analysis plays an important role in creating the lyric. This helps the system to understand the requirements of the Director / User before beginning to create the lyrics. There are certain Criteria which I have identified to understand the requirements. They are Song Situation, Charecters/Actors, Song Duration, Song Structure, and whose Point of view, Solo/Duet/Chorus, Genre, Mood, Language Style, Poetic Strength, Singer, Role of Song, Reference, Song Visuals, and Deadline.

9. Utilizing Sentence Structure Parse

Although our Quadgram model produces sentences that are usually “grammatically correct” according to rap lyrics in general, we do not actually utilize a model that models grammatical sentence structure. One thing we could have done is use a parser, such as the Stanford Parser, to create parse trees from sentences in our corpus. Because the default training data for the Parser is from a different domain, we would probably have to manually create several “gold standard” parses based on rap lyrics.

Once we generate these parse trees, we can count the part-of-speech word pairs to create probabilities of a word given a part of speech. We can also count the parent-node!child-node-listpairs. This can be useful to create trees of our own. Starting at the root, we can recursively build a tree downward randomly based on our calculated probabilities until we end up with a tree with only parts of speech at the bottom. Then, using this “madlib” structure, we can generate a word for each part of speech. This, we believe, will generate sentences with generally good grammar and good vocabulary.

10. Sentence Generation

For each section in the song (chorus or verse) we generate a set number of lines using the correspond-ing language model. For each line we generate, we actually generate a certain number of candidate lines (K, default = 30) from the model, and rank them according to a score. Then we pick the sentence with the best score, and repeat the process to generate all the lines in that section.
This score comprises of several different metrics:

- The log probability of the sentence from our language model, divided by sentence length
- The log probability of the sentence length
- The sum of logs of TFICF (term frequency-inverse corpus frequency) of each word in the sentence
- Whether the last word of the line rhymed with the last word of the previous line
- Whether the last word of the line rhymed with another word in the sentence
- Whether the last word of the line had the same number of syllables as the last word of the previous line.

11. Notes for Each metric

- We want to make sure that the generated sentence actually fit the model we were generating from, so we calculate the sentence probability based on the model.

  We had to divide the log probability of the sentence by sentence length because longer sentences have lower probabilities due to the fact that more word probabilities are being multiplied together. This takes out the bias toward shorter sentences, so then we can utilize our second metric to score based on sentence length.
- We want our sentences to emulate the length of the sentences in rap lyrics, so we tried to account for sentence length in our score. The most common sentence length was 9 for verses, and 8 for choruses.
- To include thematic information from a given input song, we generate patterns for each word in our song.

We define pattern as the probability of the word in the song divided by the probability of the word in the corpus, which corresponds to how important and specific the word is to that particular song. If a word in our generated sentence is not in our song, we defined pattern as the minimum pattern designed. So our score metric is just the sum of the logs of these patterns for each word in the generated sentence.

12. Clustering To Create Themes

One problem with our generator is that themes seem to switch from line to line. For example, one line may be about carrying weaponry, and then the next may be about going on a bike ride. Given a corpus of lyrics, we think it may be useful to cluster songs, or even lines, based on vocabulary. For each cluster, we can build a model that trains on only lines from that cluster. Then, for song generation, we can just use one of these themed models to generate lines that relate to a specific theme.

13. Structure And Mapping

This Figure presents a flower structure and the mapping of lyric features explained in the previous section to visual features of lyric.

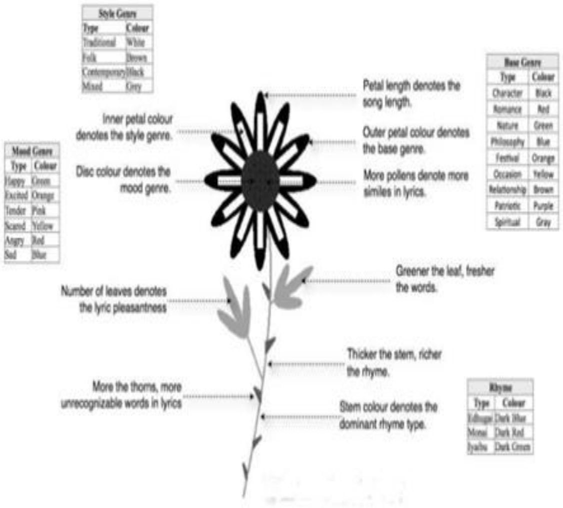
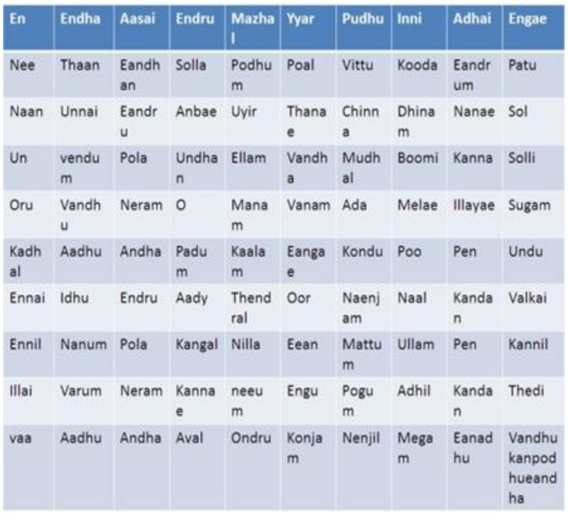

Style Genre
Type Colour
Traditional White
Folk Brown
Contemporary Black
Mixed Grey
Petal length denotes the song length.
Base Genre
Type Colour
Character Black
Romance Red
Nature Green
Philosophy Blue
Festival Orange
Occasion Yellow
Relationship Brown
Patriotic Purple
Spiritual Gray
Inner petal colour denotes the style genre.
Outer petal colour denotes the base genre.
Mood Genre
Type Colour
Happy Green
Excited Orange
Tender Pink
Scared Yellow
Angry Red
Sad Blue
Disc colour denotes the mood genre.
More pollens denote more similes in lyrics.
Greener the leaf, fresher the words.
Number of leaves denotes the lyric pleasantness
Thicker the stem, richer the rhyme.
More the thorns, more unrecognizable words in lyrics
Stem colour denotes the dominant rhyme type.
Rhyme
Type Colour
Edhugai Dark Blue
Monai Dark Red
Iyaibu Dark Green

Fig.1: Mapping Image

Table1: Mapping table



En	Endha	Aasai	Endru	Mazhal	Yyar	Pudhu	Inni	Adhai	Engae
Nee	Thaan	Eandhan	Solla	Podhum	Poal	Vittu	Kooda	Eandrum	Patu
Naan	Unnai	Eandru	Anbae	Uyir	Thanae	Chinna	Dhinam	Nanae	Sol
Un	vendum	Pola	Undhan	Ellam	Vandha	Mudhal	Boomi	Kanna	Solli
Oru	Vandhu	Neram	O	Manam	Vanam	Ada	Melae	Illayae	Sugam
Kadhal	Aadhu	Andha	Padum	Kaalam	Eangae	Kondu	Poo	Pen	Undu
Ennai	Idhu	Endru	Aady	Thendral	Oor	Naenjam	Naal	Kandan	Valkai
Ennil	Nanum	Pola	Kangal	Nilla	Eean	Mattum	Ullam	Pen	Kannil
Illai	Varum	Neram	Kannae	neeum	Engu	Pogum	Adhil	Kandan	Thedi
vaa	Aadhu	Andha	Aval	Ondru	Konjam	Nenjil	Megam	Eanadhu	Vandhukanpodhueandha

14. Conclusion And Future Work

In this paper, i have proposed a creation technique which can be used to create a lyric by using the ten prominent features of the lyric. These Features have a pattern and we can create various patterns using appropriate algorithms. It is important for us to specify the meaning and usage of a word in a computer database. The database may vary from one language to the other but the concept remains the same. This application can be used to generate lyrics to any language provided we have loaded the database with appropriate language words and the textual descriptions. In the future, I plan to find out the technique for creation of simile's and metaphors and implement in the lyric generator. This would be a challenging problem as we cannot make a machine to feel natural things such as climate, atmosphere, environment, emotions etc from which usually the poets and lyricists inspire. But it is possible to specify and categorize the words of a language into different feelings and appropriate situations.

References

[1] Automatic generation of poetry: an overview by Hugo Gonc¸alo Oliveira, Universidade de Coimbra, Portugal.

[2] Rap Lyric Generator by Hieu Nguyen, Brian Sa

[3] A Computational Approach to Re-Interpretation: Generation of Emphatic Poems Inspired by Internet Blogs by Joanna Misztaland BipinIndurkhya,

[4] Text Visualization of Song Lyrics by Jieun Oh,

[5] Lyric Visualization by Rajapandian, NandiniKarky, Elanchezhiyan, Karthikeyan&MadhanKarky,

[6] K. Binsted. Machine humour: An implemented model of puns.

[7] E Reiter, R Robertson, and LM Osman(2003). Lessons from a Failure: Generating Tailored Smoking Cessation Letters. Artificial Intelligence

[8] Al Rifaie, M., and Bishop, M. 2012.Weak vs. strong computational creativity. In Computing, Philosophy and the Question of Bio-Machine Hybrids:

[9]Bateman, S., Gutwin, C., and Nacenta, M.Seeing things in the clouds: the effect of visual features on tag cloud selections.

[10] Kikuchi.G; Lyrics-based automatic music image generation using scene knowledge for music browsing;

R. Sam Thayakaranis holding Under Graduation Degree in BCA Computer Application from SRM University,Katankulathur and pursuing Post Graduation on Master of Computer Applications from S.A Engineering College. This paper is a part of curriculum covered under in (MC7413) Technical Seminar & Report Writing
.

S.StephenJebakumaris holding Under Graduation Degree in BCA Computer Application from Jaya College of Arts and Science and pursuing Post Graduation on Master of Computer Applications from S.A Engineering College. This paper is a part of curriculum covered under in (MC7413) Technical Seminar & Report Writing.