

Lyric Generation Using Semantic Interpretation

| | | |
|-----------------|-----------------|-----------------|
| Jalin Gladis.D | Ganga.K | Dhivya Prabha.G |
| DCSE | DCSE | DCSE |
| CEG | CEG | CEG |
| Anna University | Anna University | Anna University |
| Chennai | Chennai | Chennai |

February 7, 2012

Contents

| | | |
|----------|-----------------------------------|----------|
| 1 | Abstract | 1 |
| 2 | Detailed Analysis | 2 |
| 2.1 | Morphological Analyser | 2 |
| 2.2 | Stop Words Remover | 2 |
| 2.3 | Semantic Interpreter | 2 |
| 2.4 | Accumulator | 2 |
| 2.5 | Lyric generator | 2 |
| 3 | Detailed design | 3 |
| 3.1 | Architectural Design | 3 |
| 3.2 | Data Flow Diagram | 4 |
| 4 | Techniques | 4 |
| 4.1 | Lyric model | 4 |
| 4.2 | Lyric generator | 5 |
| 4.3 | Morphological generator | 5 |
| 5 | Platforms and tools used | 5 |
| 6 | Algorithm | 5 |
| 7 | Accessing the dataset | 6 |
| 7.1 | Processing the input | 7 |
| 7.2 | Access the corpus | 7 |
| 8 | Expected outcome | 7 |

1 Abstract

The aim of the proposed system is to generate lyrics in the Tamil language automatically given a scene description. This automatic lyric generator must also

consider the factors of poetry like rhyming, alliteration, simile etc to make the generated lyrics musical. The bottleneck in the system is to generate meaningful lyrics that adhere to the given scenario and also bring in logical coherence between the lines generated.

2 Detailed Analysis

The detailed analysis of the system may be summarized as follows. A ray on the modules involved in the system

2.1 Morphological Analyser

The input to the system is a vernacular description of the scene. The morphological analyzer takes each word from the given scene description and finds the root word for these

2.2 Stop Words Remover

As soon as the root words are identified , the stop words are eliminated. That is the words which are identified as preposition, conjunction, interjection etc gets removed.

2.3 Semantic Interpreter

The semantic interpreter analyzes the words to identify the tense (present, past, future), gender (male, female), place of speech (first person, second person, third person) which are essential for generating lyrics. The system also involves finding the emotion of the scene by identifying the words which best convey the actual atmosphere of the situation . This component is very much crucial to convey the actual mood of the given situation.

2.4 Accumulator

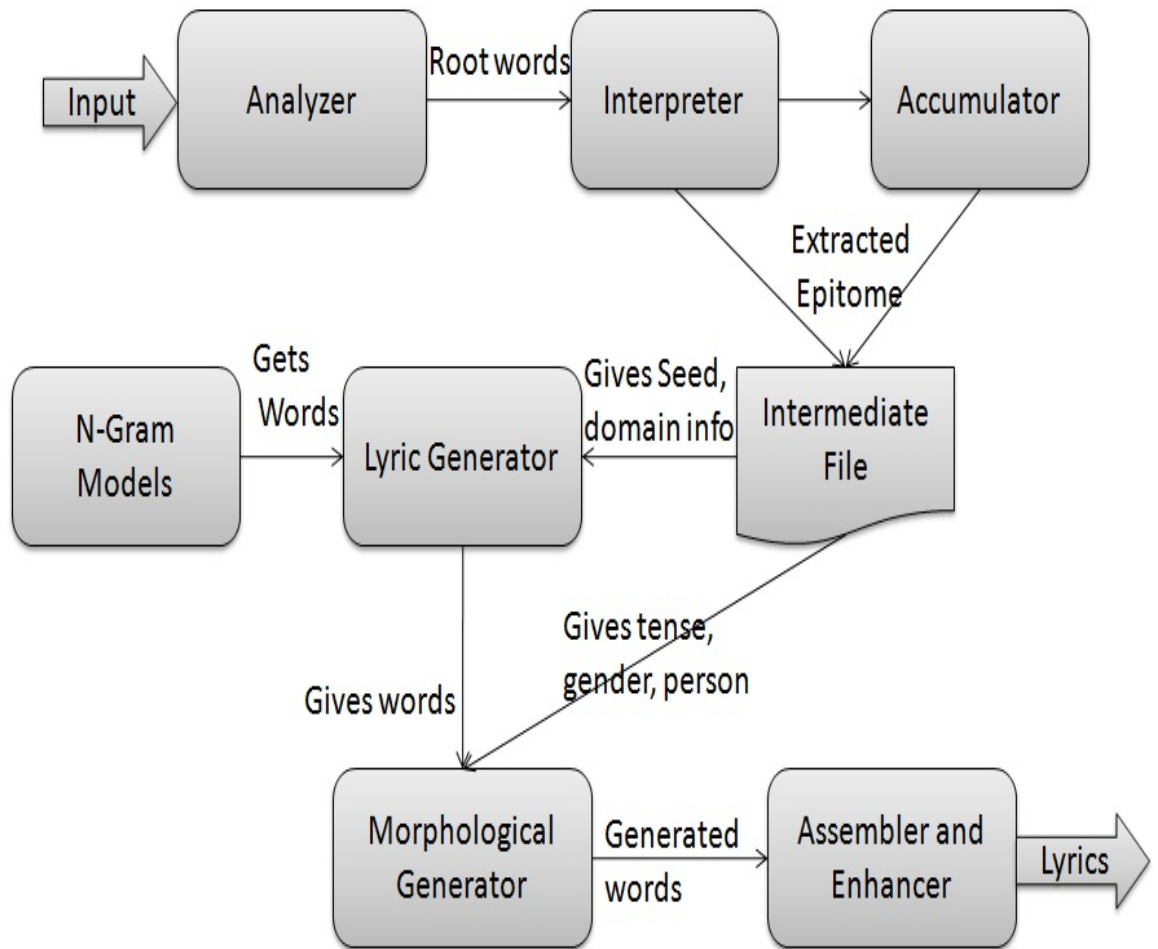
Once the seed gets generated the accumulator is triggered to gather more words that occur frequently in this context, words with similar meaning and phrases that are closely associated to the seed which are then given to the lyric generator.

2.5 Lyric generator

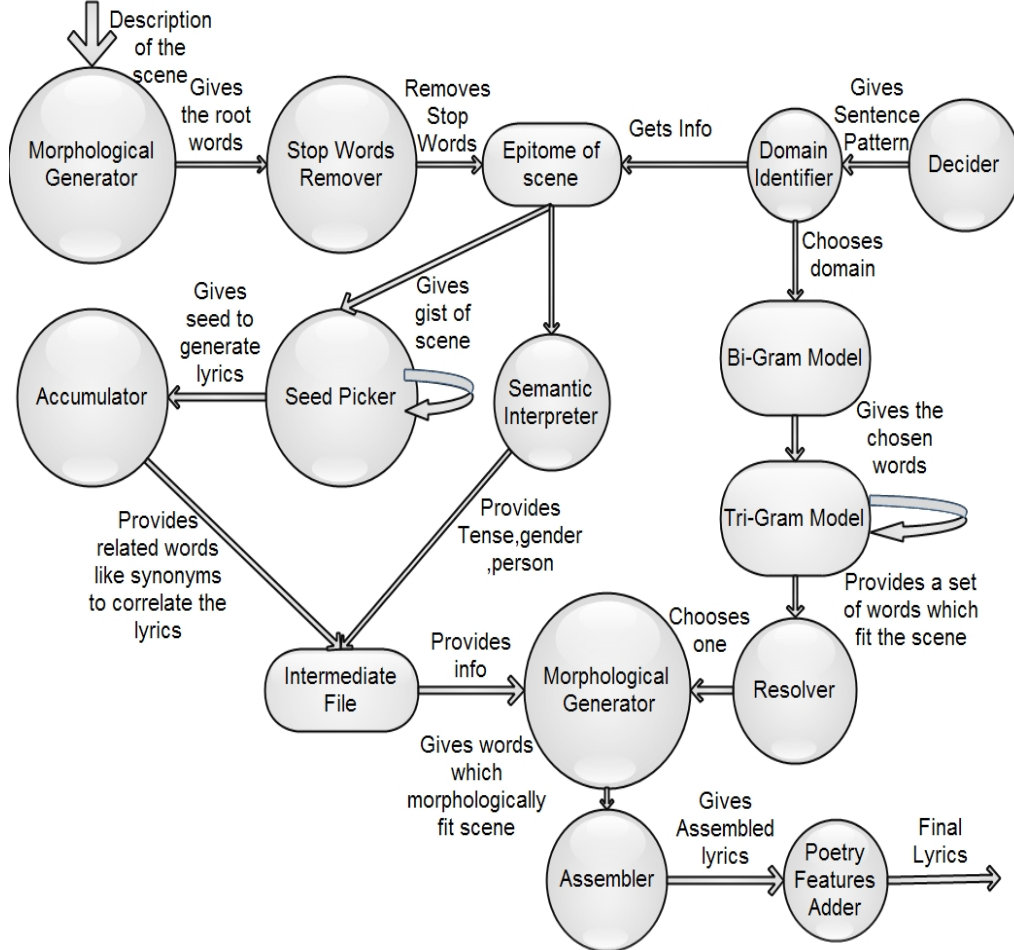
The lyric generator uses the rules of Tamil grammar and the output of the previous modules to generates lyrics. It also includes rhyming, alliteration etc to have a better feeling of the song.

3 Detailed design

3.1 Architectural Design



3.2 Data Flow Diagram



4 Techniques

The techniques used in major modules of the system can be elaborated as follows.

4.1 Lyric model

The lyric model is based on semantic interpretation. To bring this semantic correctness we propose a Tri-Gram model where the words are chosen relative to the previous ones. For better performance we opt a technique that makes use of blend of words and co-occurrences. The trigram is constructed based on a probabilistic measure to ensure that most suited words form a sentence and the correlation is maintained. A randomization technique is used to ensure that same words are not chosen always for similar situations. Thus this is randomized probabilistic trigram model.

Consider the sentence pattern N-V-AD (Noun Verb Adverb) for which the model is constructed. We design a bigram between the N- V and extend in to a trigram for N-V-AD. Initially a noun say N_i is chosen in a random manner and using the bigram model designed, we go for selecting the verb V_i , based on the probability of the occurrence of a particular verb following that noun and let this probability be P_i . We repeat the process of selecting verbs for this noun with different probabilities say $P_{i1}, P_{i2}, P_{i3}..$ Once we have chosen the nouns we go for selecting the AD_i for each $\langle N_i, V_i \rangle$ pairs based on the trigram.

4.2 Lyric generator

The lyric generator then takes the output set $\langle N_i, V_i, P_i \rangle$ and goes about computing the cumulative probability for these sets using the Bayes theorem. Out of these the pair that yields the maximum probability is selected for further processing. The words obtained are root words and hence they are sent to the morphological generator along with the details like tense marker, gender information etc that this module collects from the intermediate file. The process continues for the required number of times and the lyric generator is the one that suggests the sentence pattern to the lyric model in picking up words.

4.3 Morphological generator

The morphological generator takes the triplet $\langle N, V, AD \rangle$ and also the other required information such as tense marker, gender and the urubu and frames the sentence accordingly to the rules given. The actual working of this is based on grammatical rules of Tamil, where in the words are analyzed and based on their structures the completion of the root word to its packed form is constructed.

5 Platforms and tools used

- Net Beans IDE used for developing the system.
- Azhagi Free tamil software for Tamil transliteration.
- jexcelapi- Java Excel API is a mature, open source java API enabling to read, write, and modify Excel spreadsheets dynamically. Excel spreadsheets were used for building the corpus.

6 Algorithm

lyricgen()

1. Initialize $i=1, j=2$
2. Pick a sentence pattern say S from the set
3. $\langle A_i, A_{i+1} \rangle = extract(S, 2)$
4. $W = BiGramGen_{Ai}(A_i)$
5. $\langle C, P1 \rangle = BiGramGen_{ANexti}(A_i, A_{i+1}, W)$

6. $\langle A_{i+j} \rangle = \text{extract}(S, 1)$
7. $\langle D, P2 \rangle = \text{TriGramGen}(A_{i+j-2}, A_{i+j-1}, A_{i+j}, W, C)$
8. $W = \text{Resolver}(W, C, D, P1, P2)$
9. if(j exceeds MAXWORDS(S)) return W write to output file else Goto step 5

BiGramGen_{AI}(A_i)

1. Open file corresponding to A_i
2. Pick a random W_i
3. Add W_i to W
4. return W

BiGramGen_{ANexti}(A_i, A_{i+1}, W)

1. Open file corresponding to $\langle A_i, A_{i+1} \rangle$
2. Choose random C_i s for W_i
3. Add C_i s to Cand and the corresponding probabilities $P1_i$ s to P1
4. return C,P1

TriGramGen($A_{i+j-2}, A_{i+j-1}, A_{i+j}, W, C$)

1. Open file corresponding to $\langle A_{i+j-2}, A_{i+j-1}, A_{i+j} \rangle$
2. Pickup D_i with maximum P_i for $\langle W_{i+j-2}, C_i \rangle$
3. Add D_i s to D and the corresponding probabilities $P2_i$ s to P2
4. return D,P2

Resolver(W,C,D,P1,P2)

1. CP= Multiply $P1_i$ s from P1 with corresponding $P2_i$ s from P2
2. Get W_{i+j-1}, W_{i+j} corresponding to maximum in CP
3. Add $\langle W_{i+j-1}, W_{i+j} \rangle$ to W
4. Increment j for next iteration
5. return W

extract (S,n)

1. return n components (Noun, Verb, Adverb etc.,) of sentence for the required iteration.

7 Accessing the dataset

The data involved in the system includes the corpus of words used in generating poetry and the input text document

7.1 Processing the input

The input text document is nothing but a description of the scenario in vernacular language. It is processed to get root words of each individual word, analyzed to find the emotion and other required information for the functioning of the remaining modules.

7.2 Access the corpus

The corpus is being constructed analyzing around 150 songs in Tamil language for each emotion and identifying the words used frequently in the domain. The words and co- occurrences are stored in the excel spreadsheet , grouping them to the mood for effective usage . Access to the dataset is using the jexcel API.

8 Expected outcome

The input to the system will be a text document which is a vernacular description of the scene for which the lyrics must be generated. The output will be a text document with lyrics. The project will be demonstrated for different inputs which would include 2-3 domains like love, melancholy, friendship etc., Parameters like relatedness to the context, number of lines generated, rhyming and other poetic features like alliteration, simile etc must be taken care of.

References

- [1] Ananth Ramakrishnan A, Sobha Lalitha Devi ,June 2010 *An alternate approach to meaningful lyric generation in Tamil* , Workshop on computational approaches to linguistic creativity, Los Angeles, California
- [2] Burr Settles,June 2010 *Computational creativity tools for song writers* , Workshop on computational approaches to linguistic creativity, Los Angeles , California
- [3] Karthika Ranganathan, T.V Geetha, Ranjani Parthasarathi, Madhan Karky,2011 *Lyric Mining: Word, Rhyme and Concept Co-occurrence Analysis*,Tamil Internet
- [4] S. Lakshmana Pandian, T.V.Geetha,2009 *Semantic Role based Tamil Sentence Generator*, International Conference on Asian Languages Processing