Exploring different strategies for the automatic generation of song lyrics with Tra-la-Lyrics

Hugo R. Gonçalo Oliveira F. Amílcar Cardoso and Francisco C. Pereira
{hroliv, amilcar, camara}@dei.uc.pt

CISUC, Dep. of Informatics Engineering, University of Coimbra, Portugal

Abstract. Using as starting point the architecture of a system that generates lyrics for given melodies, this paper explores possible different strategies for their automatic generation. Some important features concerning lyrics generation and some problems about different strategies are introduced. Examples of generated lyrics are shown and discussed. The results were validated and evaluated. People were invited to answer evaluation inquiries where some lyrics could be found. The evaluation results are also a topic for discussion.

1 Introduction

In the last few years we progressively accepting the computer as a an artificial artist, highly contributing for turning the creative systems an interesting topic for research. Composing musical pieces [1, 2], telling stories [3, 4], making up jokes [5, 6], creating visual art [7] or making up poetry [8–10] are just examples of creative outputs. During these later years we have also seen an increasing number of systems where users can customize digital and multimedia contents they can share with the whole community or simply keep for their own use.

We have recently started to develop a system capable of generating lyrics: Tra-la-Lyrics. Its architecture was designed with the aim of supporting the integration and subsequent testing of different generation strategies. Based on some conclusions about the relations between lyrics and rhythm (discussed in [11, 12]), strategies to get the words have been tested, in order to find out how far a system like this can go. Lyrics should obey the melody's rhythm but the choice of the words is also very important. Words' sequence should employ some features to make the lyrics interesting.

This paper starts by introducing the architecture used but it is mainly focused on the strategies used to get words, that take advantage of features also described. Some experimental demos are shown and analyzed. Results validation and evaluation are finally discussed.

2 Architecture

The architecture of Tra-la-Lyrics has two core modules: the *Metrics* module, responsible for rhythm constraints and the *Vocabulary* module that adds more constraints to the words being sought, in order to make the lyrics more interesting. The latter module can be the target of different reimplementations that we have called **strategies**. Figure 1 shows this part of the architecture.

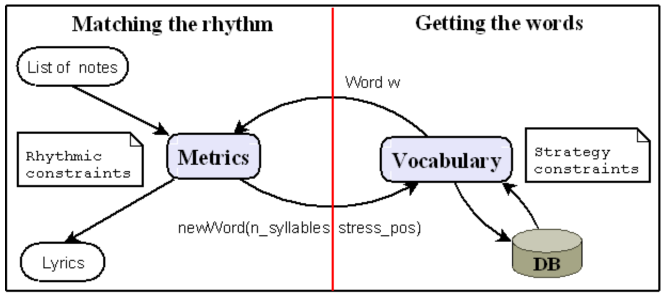


Fig. 1. Architectural separation between the algorithm to match the rhythm and the strategy to get the words.

The *Metrics* module defines a number of syllables and a stress position for each word, so that it fits the rhythm. These values are defined by **special rhythmic patterns** in the melody [11]:

- **Strong beat**: A note occurs in a strong beat.
- **Strong beat followed by a rest**: A note occurs in a strong beat and we can find a rest after it.
- **Strong beat followed by the end of the melody**: The note occurs in a strong beat and it's the last note of the melody.
- **Rest**: A rest.
- **Strong beat followed by some note, followed by a rest**: The note is a strong beat, followed by some other note, after which we can find a rest
- **Last strong beat of the current part**: The note is in the last strong beat of the current part of the music.
- **End of the melody**: The note is the last of the melody.

The *Vocabulary* module is the one we will focus on this paper. It adds some more constraints to the wanted words and then tries to get them from a database. The database was built using mainly words from the tree bank Floresta Sintctica which can be found in Linguateca[1]. It also contains words taken from Portuguese poetry. The words grammatical and morphological attributes were filled using a morphological analyzer called Jspell [13]. The syllabic related attributes were obtained using algorithms able to accomplish syllabic division and identification of the stress of a word described in Oliveira's dissertation [12]. The database structure is shown in Figure 2.

[1] http://www.linguateca.pt

Words	Morphology	Verbs	PersProns	Syllables
key	key	key	key	word
word	gender	transitive	person	n_syllables
category	number	tense	type	division
root		person		stress_pos
				termination
				rhyme_coeficient

Fig. 2. Our database's structure.

When getting the words from the database, the given number of syllabes is not handled as an exact number of syllables, but as a maximum number. If not, lyrics would tend to have only one word for each pattern. Words would be longer and only patterns with only one or two notes (very uncommon) would select shorter words.

3 Features

Manurung [8] identifies various aspects that are considered to be defining features that make up the look and sound of poetry, setting them clearly apart of other types of texts. The three identified aspects are, respectively:

- Rhythm and meter
- Rhyme and other phonetic patterns
- Figurative language

Like poetry, lyrics usually take advantage of these features. In *How to Write Lyrics*[2] it is suggested that there are three important points one should pay attention when choosing words for a song: **rhythm**, **rhyme** and **repetition**.

As said in the Section 2, rhythm is handled by the *Metrics* module. A strategy deals with the other features in different ways, depending on its implementation.

3.1 Rhymes

We consider that two words rhyme with each other when they have the same termination. To obtain the words' terminations there are three simple steps:

1. Identification of the word's stressed syllable.
2. Identification of the most stressed vowel
3. Detachment of all the characters of the word, starting on the most stressed vowel till the end of the word.

Word terminations are stored in our database.

Although we are planning to implement an automatic way of identifying suitable places to have rhymes with an approach similar to the one of Grilo [14], at the moment it is only possible to input a possible division of the music into phrases. The implemented strategies try to have rhymes at the end of each one of them.

[2] http://everything2.com/index.pl?node=How%20to%20write%20lyrics

3.2 Repetition

In order to have some repetition of sound in our lyrics the implemented strategies can have some of the following features:

1. **Word reuse**: every word selected to be part of the lyrics is stored. Since then, every time a new word is needed, if any of the previously chosen words suit the actual constraints, there is a probability of reusing it. This probability can be changed.
2. **Words with the same root**: The user can input a list of open class words[3] he would like to see in the resulting lyric. The system will then give priority to words with the same root as the ones in that list. We also call this the *subject* feature because it is an attempt to make the lyrics have some subject. The roots of the words are stored in our database.
3. **Parallelism between sentences and musical phrases**: This feature strengthens both the notion of rhythm and the present repetition. If each sentence in the lyrics corresponds to a fixed number of musical phrases, they will be easier to sing and will have a more logical structure, since lyrics will be more strictly connected to the music. To accomplish this, we force each sentence to end in the last note of the group of musical phrases, matching their ends. If it simply doesn't make sense to end the sentence with a word belonging to the class of the last word[4], that word is deleted. Then either an interjection is selected to replace the deleted word or the last syllable of the previous word is strained till the end of the musical phrase. The number of musical phrases for each sentence can be changed.

4 Strategies

In [11] we suggest three possible strategies to add constraints to the *Vocabulary* module in order to generate more interesting lyrics. Two of the proposed strategies have been tested (Random words and grammar) and we have introduced a new one (Generate and test) with a completely different approach from the others. In this Section, the three strategies are described.

4.1 Random words

Strategy description: This is the most simple implemented strategy. It basically tries to get random words that obey the constraints given by the *Metrics* module, in order to fit in the rhythmic pattern. This strategy hasn't got any notion of sentence and each word is independent from the other. To make the lyrics generated by this strategy a little bit more interesting we have added two of the features, described in Section 3: **rhymes** and **word reuse**.

[3] Nouns, verbs and adjectives

[4] If the word is an article or a preposition it shouldn't be the last one of a sentence

Priorities: Although we have a big database and this strategy has few constraints to get the words, sometimes there aren't suitable words at all. That is why we have defined a list with the priorities each constraint has (Table 1). In our opinion rhymes are very important and there is no problem if we get a little amount of syllables that not match the rhythm. In a preliminary study [12] some unstressed syllables were found in strong beats. This tells us there is no problem about letting the stress position being the first constraint to fall. If there is still no word, rhythmic constraints are recovered, and termination constraint falls.

Rhythm constraints	**Strategy constraints**
max syllables + (stress position or no stress)	termination
max syllables + min/max stress position	termination
max syllables + (stress position or no stress)	none
max syllables + min/max stress position	none

Table 1. Constraint priorities in the random words strategy

4.2 Generative grammar

Strategy description: This is the strategy we are currently developing. It still includes some constraints related with rhymes and repetition but it has a generative grammar that builds Portuguese sentence templates. Every time a new word is needed, it should agree not only with rhythmic constraints but also with the ones given by the next grammatical category in the actual template. This strategy takes advantage of all the three features related with repetition that were earlier described (Section 3): **word reuse**, **words with the same root** and **parallelism between sentences and musical phrases**. There is however a difference present in the words reuse process: only open class words are kept for future reuse.

The grammar: The grammar was implemented by ourselves and generates sentence templates. Each template is basically a list filled with symbol instances. A symbol represents a grammatical category with its own morphological attributes. For example, each noun has a gender and a number and each verb has a number and a person. The grammar itself guarantees morphological coherence between each symbol. Figure 3 shows our grammar's productions. The meaning of each symbol can be found in Table 2. It is possible to give each production a different probability of being selected.

Avoiding the lack of rhymes: As one can notice there is a huge amount of constraints in this strategy, making it harder to find words that rhyme at the end of each musical phrase. In order to minimize the lack of rhymes, we have added the following during the selection of the last word of a part:

Symbol	Class
S	sentence
SN	noun phrase
SV	verb phrase
SA	adjective phrase
SP	preposition phrase

Symbol	Class
adj	adjective
adv	adverb
art	article
con	conjunction
n	noun
pdem	demonstrative pronoun
ppos	possessive pronoun
ppes	personal pronoun
prep	preposition
v	verb

Table 2. Symbols used and their corresponding grammatical group/category

```
S → SN SV | SN SV con | SV SP | SV SP con
SN → art n | pdem n | art n SA | pdem n SA | art SA n | art SA n SA
SV → v | v SN | v v[5]
SA → adj | ppos | ppos adj | adv adj | adj adv
SP → prep SN | adv
```

Fig. 3. Generative grammar productions

- If the sentence is the first of the lyrics or if the two sentences before rhyme with each other, a suitable word with a high rhyme coefficient[6] is selected.
- If the sentence should end in rhyme, a word whose termination is in the list is looked for to end up the sentence. The last stored termination (which was the last one used) has priority over the others.

Priorities: This strategy has to deal with many constraints: rhythmic, grammatical category, morphological and sometimes even word root or termination. We have defined a list of priorities, similar to the one in the strategy before, but longer. This time there are three different combinations of constraints:

1. **Simple case**: word with rhythmic, grammatical and morphological constraints.
2. **Simple + rhymes**: word with rhythmic, grammatical, morphological and termination constraints.
3. **Simple + root**: word with rhythmic, grammatical, morphological and root constraints.

Grammatical and morphological constraints can never fall, because that could lead to inconsistent sentences. Like in the previous strategy, the stress position is the first constraint to get rid of. As an example, priorities for situation 2 are shown in Table 3. Priorities for situation 3 would be similar, but with a word root constraint instead of termination.

[6] The rhyme coefficient is stored in our database and is proportional to the number of words with the same termination.

[7] Except if the stress is in the first syllable or if there shouldn't be stress at all. If this is the case, stress position constraint falls.

Rhythm constraints	**Strategy constraints**
max syllables + (stress position or no stress)	grammatical and morphological attribs + termination n
max syllables + min/max stress position[7]	grammatical and morphological attribs + termination n
max syllables + (stress position or no stress)	grammatical and morphological attribs + termination n-1
max syllables + min/max stress position[7]	grammatical and morphological attribs + termination n-1
...	...
max syllables + (stress position or no stress)	grammatical and morphological attribs
max syllables + min/max stress position)	grammatical and morphological attribs

Table 3. Constraint priorities in the generative grammar strategy, when we want to have a rhyme

If the end of the priorities list is reached and a word wasn't found the grammar can advance to next grammatical category if the actual was defined as optional[8]. If not, backtracking is used. The last word is deleted and another word with the same category is looked for. This happens until a word is found or a new template is selected.

4.3 Generate and test

Strategy description: This strategy follows a completely different approach from the other two. It uses a generate and test algorithm to select the best possible sentences for each musical phrase. Each complete sentence is generated by the grammar and is then evaluated. If its score is higher than the previous lowest, nothing happens. Otherwise it replaces the previous lowest. As long as no sentence gets a score considered acceptable and the number predefined as the maximum generations isn't reached, new sentences are generated and evaluated.

The lyrics generation takes much longer with this strategy than with the others, rhythm is usually least matched but parallelism between music and lyrics is always present.

Evaluation function: The evaluation function is somehow inspired by the metre evaluation used by Manurung [8]. To evaluate each sentence, each candidate sentence is compared with a target metric pattern. The evaluation score is equal to the sum of all the penalties. We have defined three major kinds of penalties:

- **Size:** The sentence has a number of syllables different from the number of notes.
- **Rhythm:** There are unstressed syllables in strong beats.

[8] There are words belonging grammatical categories that can be dropped without affecting the grammatical correctness (adjectives, for example)

– **Sound:** There are no rhymes at the end of the sentences or there are words split by a rest.

5 Demo runs

In this section some music with respective generated lyrics will be shown. Lyrics generated by strategies with grammatical notions have the words' grammatical categories just below them.

5.1 Random words

The music shown in Figure 4 is from the portuguese song for children called *Três Pombinhas*. The generated words sequence is random and rhythm is perfectly matched. The probability for word reuse was set to 80% making the words *provisórios* being repeated several times. Musical phrases end up with the rhymes *servirão/expansão* and *pilar/andar* respectively.

Vigiava inflamáveis continentes servirão
passaporte provisórios tiraram expansão
provisórios brilhante provisórios pilar
provisórios desmente provisórios andar

Fig. 4. Lyrics generated with the Random Words strategy

5.2 Generative grammar with given words

The music shown in Figure 5 is from the portuguese song for children called *O Barquinho*. The generated lyrics agree with grammatical rules and rhythm is matched (*sentindo* is the only word not matching). The probability for word reuse was set to 30% but no words were reused. There are three words that rhyme: *carvão*, *alemão* and *betão*. Words like *rosa*, *sentiu*, *amado*, *ame*, *sentem*, *sentindo*, *amor*, *amante*, *sinta* and *sentirá* have the same root as the words given as subject, which were, respectively: *sentir*, *amor*, *amar* and *rosa*.

Uma rosa sentiu a secção sentiu ali um amado carvão
ame esses amores sedados sentem sentindo o alemão
alerta amor amante bastante sinta jamais sentirá tal betão
art n v art n . v adv . art adj n
v . pdem n adj v v . art n
v . n adj adv v adv . v . pdem n

Fig. 5. Lyrics generated with the Generative Grammar strategy

5.3 Generate and test

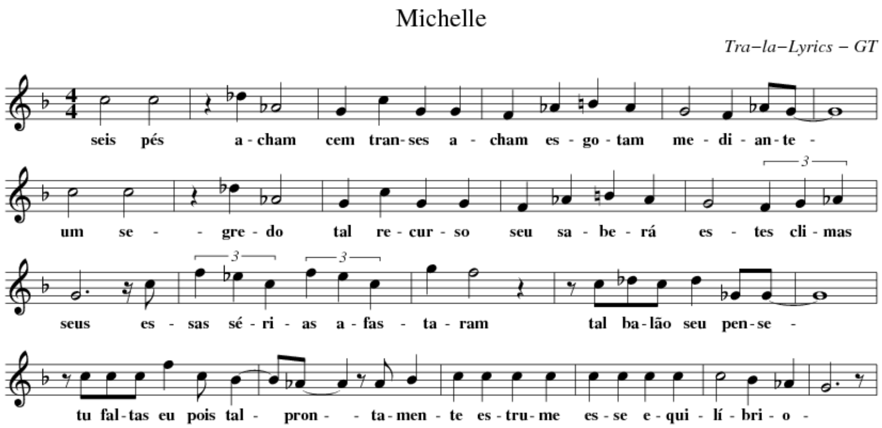
The music shown in Figure 6 is from a well known song by *The Beatles*, *Michelle*. The generated lyrics have some words that don't match the rhythm like *acham*, *mediante* or *prontamente*. There are no rhymes, but the word sequence agrees with grammatical rules. There is also a parallelism between the musical and the lyrics structure, because each musical phrase is associated with one sentence.

6 Validation and evaluation of the results

6.1 Validation

In Oliveira's dissertation [12] some results about the rhythm in 42 portuguese songs were published[9]. We selected three melodies with different metre types and thirty lyrics were generated to each one of them. This was done for the

[9] Mainly correlations between syllable and beat strength



Seis pés acham cem transes acham esgotam mediante
um segredo tal recurso seu saberá estes climas
seus essas sérias afastaram tal balão seu pense
tu faltas eu pois tal prontamente estrume esse equilíbrio
art n v . art n v . v prep
art n . pdem n ppos v pdem n
ppos . pdem n v . pdem n ppos v .
ppes v ppes con . pdem n v pdem n

Fig. 6. Lyrics generated with the Generate and Test strategy

three strategies, using the same melodies. The rhythm of the generated lyrics was analyzed and our results were in agreement with the ones in the "real" songs. The lyrics generated by the Random Words and Generative Grammar strategies had a significantly lower amount of notes not matching the rhythm.

6.2 Evaluation

As we all know, creative systems are not easy to evaluate since their quality tends to be very subjective. Requesting human opinions is one of the few possible ways of creative outputs and that is how many authors, like Kim Binsted [5] or Federico Peinado [15] have evaluated their own system. We have made an online evaluation inquiry during two weeks that was answered by 70 people.

The inquiries had four parts, each one with different generated lyrics for the same song (portuguese children's song *Papagaio Louro*) in sheet music format and an audible interpretation. Each part had the same questions and each one of the lyrics had beed generated using a different strategy[10]. The latter wasn't told to the volunteers. Table 4 shows the evaluated topics and the respective questions:

A large majority of rhythm ratings was positive. Lyrics generated with Random Words strategy got the better rhythm, rhymes and sound rating. The others

[10] One of the lyrics had been generated by a strategy not described in this paper.

Topic	Questions
Rhythm	Quality rating and sentences identification.
Rhymes	Quality rating and choice of the better ones.
Sound	Quality rating and choice of words that sound better/worse.
Semantics	Meaningful rating and a choice of a suitable title
Quality	Overall quality and entertainment ratings

Table 4. Topics evaluated in the inquiries.

also got mainly positive rhythm and sound ratings but negative rhyme ratings (even though, not very far from 50%). When it comes to semantics, no lyrics were positively rated has having an explicit meaning. The Generative grammar strategy got the better ratings (42%) and the Generate and Test strategy came in second (29%). The Random Words strategy was far from those values. An interesting thing is that, even though semantics was rated very low, almost everybody was able to find a suitable title for the lyrics. 95% of the lyrics were given a title, often with with words from the lyrics or related to them. For the generation of the lyrics with the Generative grammar strategy the word *saudade* was given as subject and its plural, *saudades* appeared many times. 63% of the given chosen titles for these lyrics had one of the words. Overall quality ratings were also negative for all the strategies. Their order was the same as the semantics but the rates were closer, respectively 46%, 41% and 38%. Entertainment ratings were all above 50%, however we were expecting better results on this. The Random Words strategy got the better ratings, Generate and Test got second and Generative grammar got third, all very close (59%, 58%, 53%).

To sum it all up, the Random Words strategy generates the lyrics that better fit the rhythm, with better rhymes and sound. It also seems to generate funnier lyrics, but often nonsense. Although none of the strategies have semantic notions, we can conclude that, if grammatical rules are obeyed, it may become easier to build a meaning. Even though the quality of the lyrics is not the best, they can be used as a piece of entertainment.

7 Conclusions and further work

Although, according to the inquiries, some interesting lyrics were generated by our strategies, they are still far from the ones a human can write. In order to improve this behavior we are planning to keep working in the generative grammar strategy. The number of production rules in the grammar might be enough but it should have more specific rules if we want it generate more natural sentences. Some heuristics to improve matching the end of the sentences in the lyrics with the end of the musical phrases can be defined. The template selection could be based on the prediction of the average number of syllables each template can have. More strategies to get the words can easily be integrated in the actual system. Possible strategies like one with a semantic module or even a completely evolutionary approach could be tested. Another really interesting experiment would be using sung voice synthesis software like Singing Computer[11] to sing our generated lyrics.

References

1. Cope, D.: An expert system for computer-assisted music composition. Computer Music Journal 11,4 (Winter) (1987)
2. Todd, P.M., Werner, G.N.: Frankensteinian methods for evolutionary music composition. Musical networks, MIT Press (1999) 313–339
3. Bringsjord, S., Ferrucci, D.A.: Artificial Intelligence and Literary Creativity: Inside the Mind of BRUTUS, a Storytelling Machine. Lawrence Erlbaum Associates, Hillsdale, NJ. (1999)
4. Gervás, P., Lönneker-Rodman, B., Meister, J.C., Peinado, F.: Narrative models: Narratology meets artificial intelligence. In: Basili, Roberto and Lenci, Alessandro (Ed.). International Conference on Language Resources and Evaluation. Satellite Workshop: Toward Computational Models of Literary Analysis, Genova, Italy. (2006)
5. Binsted, K.: Machine humour: An implemented model of puns. PhD thesis, University of Edinburgh, Scotland (1996)
6. Ritchie, G., Manurung, R., Pain, H., Waller, A., Black, R., O'Mara, D.: A practical application of computational humour. In: Cardoso, A. & Wiggins, G. (Ed.). Proceedings of the 4th. International Joint Workshop on Computational Creativity, London, UK. (2007)
7. Machado, P., Cardoso, A.: Nevar - the assessment of an evolutionary art tool. In: Wiggins, G. (Ed.). Proceedings of the AISB00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science, Birmingham, UK. (2000)
8. Manurung, H.: An evolutionary algorithm approach to poetry generation. PhD thesis, University of Edinburgh (2004)
9. Gervás, P.: Wasp: Evaluation of different strategies for the automatic generation of spanish verse. In: Wiggins, G. (Ed.). Proceedings of the AISB00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science, Birmingham, UK. (2000)
10. Gervás, P.: An expert system for the composition of formal spanish poetry. Journal of Knowledge-Based Systems **14** (2001) 181–188
11. Oliveira, H.R.G., Cardoso, F.A., Pereira, F.C.: Tra-la-lyrics: an approach to generate text based on rhythm. In: Cardoso, A. & Wiggins, G. (Ed.). Proceedings of the 4th. International Joint Workshop on Computational Creativity, London, UK. (2007)
12. Oliveira, H.R.G.: Geração de texto com base em ritmo. Master's thesis, University of Coimbra (2007)
13. de Almeida, J.J.D., Pinto, U.: Jspell - um módulo para a análise léxica genérica de linguagem natural. In: Actas do X Encontro da Associação Portuguesa de Linguística. (1995) `http://natura.di.uminho.pt/~jj/pln/jspell.ps.gz`.
14. Grilo, C.F.A.: Aplicação de Algoritmos Evolucionários à Extracção de Padrões Musicais. PhD thesis, University of Coimbra (2002)
15. Peinado, F., Gervás, P.: Evaluation of automatic generation of basic stories. New Generation Computing, Computational Paradigms and Computational Intelligence. Special issue: Computational Creativity **24**(3) (2006) 289–302

[11] http://freebsoft.org/singing-computer