Komposer – Automated Musical Note Generation based on Lyrics with Recurrent Neural Networks

Dulan. S. Dias[#1], T. G. I. Fernando[#2]

[#]*Department of Computer Science, University of Sri Jayewardenepura*

Gangodawila, 10250, Sri Lanka

[1]dulan@ieee.org

[2]gishantha@dscs.sjp.ac.lk

***Abstract*— Musical creativity being one of the strong-hold characteristics that differentiate humans from computers in today's technologically advanced society, algorithmic composition and song writing are the research areas that aim to bridge this gap. With the introduction and development of various neural network-based methodologies that have shown quite a promise in applications to a wide range other fields, it is promising to see how these new technologies can cater to the domain of musical creativity. Even though there has been significant amount of research done focusing on musical composition, it is not the same for musical song writing. The main objective of this research study is to apply Long Short-Term Memory Recurrent Neural Networks in constructing a machine learning model that can generate musical melody notes when it is provided with a lyrical input (musical song writing). In this study, we were able to successfully generate musical melody notes for provided lyrical inputs with consistencies of over 80%. In addition to that, a web-based inference tool was developed as a result of this study, which allows us to easily generate musical melody sheets when we provide with a lyrical input.**

***Keywords* — musical song writing; recurrent neural networks; lyrics; musical note generation.**

I. Introduction

Music is an art form which is believed to have originated in the Palaeolithic era of the human history from various parts of the world. Music is merely sound organized in time, but which differ according to social and cultural context in the creation, performance, significance and even in the definition of music itself. Any musical piece could be characterized based on its pitch, rhythm, dynamics and the sonic qualities of timbre and texture. Music is an art that is performed with a vast variety of instruments and vocal techniques ranging from singing to even rapping; there are solely instrumental pieces, solely vocal pieces and pieces that combine both singing and instruments together. Music can be divided into genres (e.g., country, pop, hip-pop, rock) and genres can be further divided into sub-genres (e.g., country blues and pop country), although the relationships between music genres are often subtle, it varies based on personal interpretation, and sometimes controversial as well.

In many cultures, music has become an important part of people's day-to-day life, since it plays a key role in religious rituals, rite of passage ceremonies (e.g., graduation and marriage), social activities (e.g., dancing) and cultural activities ranging from amateur singing to playing in an amateur rock band or singing at a church or school choir. Music may be made as a hobby, like a student playing a guitar in the college band, or work as a professional musician or a singer. The music industry includes individuals who create new songs and musical pieces (such as songwriters and composers), individuals who perform music (which include orchestra and band, musicians and singers), individuals who record music (music-producers and sound engineers), individuals who organize concerts and tours, and individuals who sell recordings of music [1].

Production of musical pieces has traditionally relied on a wide range of specialized expertise far exceeding the capabilities of most single individuals. Recent advancements in music technology and the parallel development of electronic music have brought the joy of music-making to a greater audience. For the amateurs who enjoy expressing their creativity through music-making, there has emerged a newly-levelled playing field along with professional and semi-professional musicians who are able to independently produce music with no more than a personal computer [2].

In the simplest production of a musical piece that combines singing and instruments, the two most important components are the lyrics and melody. The recent technological advancements have made it possible for amateurs to produce melodies without needing any knowledge or skill in handling any musical instrument, with software such as 'Logic Pro,' 'Live' and even with the freely available 'Garage Band' for Mac operating systems. These tools allow anyone with access to a personal computer to create melodies by mixing different sound tracks with sound identical to that produced by musical instruments. Even with the above listed tools for melody production, human intervention is required throughout the process due to the need of creativeness of the human mind, or in the context of music, the musical ear of a human. Writing of lyrics still remains a task that needs the creative mind of a human, even though a wide amount of research is being carried out at present in the field of automated generation of original lyrics.

The aim of this research is to gap the bridge between the need for human intervention to bring forth creative musical melodies. In this research, it is targeted to develop a tool that takes the lyrics of an English song in order to produce a matching melody for that particular song by generating a sequence of musical notes for the melody. However, it is not a target of this research to produce the complete melody which includes determining the timing at which musical notes should be played and determining overall

978-1-7281-3041-5/19/$31.00 ©2019 IEEE.

characteristics of the melody such as its genre from the provided lyrics.

II. Literature Survey

Before moving on to our methodology, it was very important that we studied what the existing methodologies were and on what tools and technologies they were based on. It was also important that we understand the basics of music that will be involved in the scope of our research.

In reading deep into the terminology that have been used in similar research work, it was found that there is a huge misconception between the meanings of the terms ‘Musical Composition’ and ‘Musical Song writing.’ Before we proceed let us first clearly understand what these two terms exactly mean.

Composing and song writing are both processes of creating new music. They are essentially the same thing, but composition tends to mean production of the instrumental portion, while song writing is more contemporary music with the instrumental portion accompanied with lyrics.

The concept of algorithmic composition of music is not something new, with Pythagoras (around 500 B.C.) believing that music and mathematics were not separate studies. Bruce L. Jacob thinks that, it can be thought of as a compositional tool that simply makes the work of composers more efficient [3]. Hiller and Isaacson were probably the first who proposed a computational model which used random number generators and Markov chains for algorithmic composition [4]. Many researchers, since then, have tried to address the problem of algorithmic composition from different approaches.

Unlike the concept of algorithmic composition, algorithmic song writing is relatively new. Algorithmic song writing incorporates a lyrical factor to algorithmic composition, and hence, algorithmic song writing is the proper terminology that is related to our research work.

Even though our focus is on Musical Song writing, the approaches, methodologies, tools and technologies used in research related to Musical Composition, seems to be quite similar. Therefore, in our Literature Review, we will look into the research work done under Musical Composition as well.

A. Research around Musical Composition

According to Cope, algorithmic composition could be described as a sequence set of rules (instructions or operations) for solving (accomplishing) a particular problem (task) in a finite number of steps by combining musical parts (things or elements) into a whole composition [5].

Below we have categorized different methodologies based on their most prominent architecture:

- Random generators
- Grammar-based
- Mathematical models
- Knowledge-based systems
- Machine learning focused
- Evolutionary models
- Hybrid (mixed) systems

The categorization is not straightforward since many of the approaches are overlapping with one or another category. For example, Experiments in Musical Intelligence (EMI) [6] is a project that is categorized below as a grammar-based model, while it also can be seen as a knowledge-based system or even a machine learning focused system.

Musikalisches Wrfelspiel (a musical dice game) was a game that used a dice to randomly generate music from pre-composed set of options [7]. The most distinct problem with the random generator approach is that the quality of the composition is mostly weighted to chance and therefore, they are not able to generate consistent yet improvised compositions.

Steedman believes that the idea that there is grammar in music is probably as old as the idea of grammar itself [8]. Cope worked with algorithmic composition by combining Markov chains and other techniques (musical grammars and combinatorics) into a semi-automatic system, which he called as Experiments in Musical Intelligence (EMI), or Emmy [6]. Among other approaches using a grammar-based approach is Johnson-Liard, who also used grammar for the generation of Jazz chord progressions and bass line improvisations [9]. However, grammar-based approaches have a set of drawbacks as identified below:

- Grammar-based approaches create a hierarchical structure while most music is not. Therefore, ambiguity is necessary to allow improvisation.
- Grammar implementations do not make any claim on the semantics of the piece.
- Even though a grammar can generate a large number of musical strings, the qualities are questionable.
- Parsing of grammar is computationally expensive, especially if we try to cope with ambiguity.

It can be seen that both stochastic processes and Markov chains have been used extensively in the past for algorithmic composition, due to their low computational complexity which also makes them ideal candidates for real-time applications. Taking the motivations behind the dice game a further, Markov chains had been built from existing material and by encoding the variation in probabilities with respect to context. Some work based on mathematical models include the work of Cybernetic Composer proposed by Ames and Domino [10], Analogique by Iannis Xenakis [11], Pressing [12], Herman [13] and Harley [14], Gogins [15] and Conklin and Witten [16]. However, mathematical model based approaches fail since, Probability values must be assigned initially by analysing a large number of pieces, especially if we want to simulate one style, and since it is difficult to capture higher or more abstract levels of music.

In most Artificial Intelligence systems, having one or more Knowledge-based System (KBS) as a subsystem along with other subsystems is quite common. A KBS is built around rules and facts which are used in making decisions that can be reasoned by the system. A system such as this, looks promising because of its ability to justify the generation of composition. Ebcioglu implemented his own Back-tracking Specification Language (BSL) and this was then used to develop CHORAL, a rule-based expert system for the harmonization of chorales in the style of J. S. Bach [17]. Tsang and Aitken [18] and Pachet and Roy [19], used constraint logic programming (CLP) and constraint satisfaction techniques (CSP) respectively for harmonization, with the former being much more efficient. Some other work based on knowledge-based systems included the work of Ramalho and Ganascia [20], Zimermann [21] and Robertson et al. [22]. Reasons such as Knowledge (facts and rules) elicitation been difficult and time consuming in the domain of music, and high dependency on a human expert to find a flexible representation, can be identified as some of the drawbacks of the knowledge-based approach.

Machine Learning focused systems are systems that learn music initially, without having any prior knowledge, rules or constraints been fed in to them. These systems learn the features of the compositions provided in the training phase and use them to understand the technicalities of the construction of the compositions. These systems paved the path to rapid growth in the growth of automated musical composers. Artificial Neural Networks (ANNs) have been used extensively in the last couple of decades for musical applications by Todd and Loy [23], Leman [24] and Grinn and [25] and have been relatively successful. Some other works that used ANNs were done by Todd [26], Mozer [27], Bellgard and Tsang [28], Toiviainen [29], Hornel and Degenhardt [30], Hornel [31] and Melo [32].

Markov chains that are trained on a given dataset of compositions can only produce sub-sequences that also exist in the original dataset. Recurrent Neural Networks (RNNs) attempt to extrapolate beyond those exact sub-sequences. First attempts to generate music with RNNs, was developed by Todd, Mozer and et al., were limited by their short-term coherence [23, 25]. However, Doug was able to tackle this problem by switching from standard RNN cells to Long Short-Term Memory (LSTMs) cells. He concludes pointing out that the LSTM was able to play the blues with good timing and proper structure as long as one was willing to listen [33].

Evolutionary models based approaches include Genetic Algorithms that have proven to be very efficient search methods especially in very large search spaces and effective because of their ability to provide multiple solutions [34, 35].

Hybrid systems are ones which use a combination of different AI techniques and is very simple and logical. Since different AI methods come with different strengths we can adopt a postmodern attitude [36] by combining them. The main disadvantage of hybrid systems is that implementation, verification and validation are time-consuming.

B. Research around Musical Song writing

Finnish art songs were generated using a system, M.U. Sucus-Apparatusf [37] by Toivanen et al. Here, rhythm patterns were randomly chosen from among those usually found in art songs. Then, chord progressions were subsequently generated by using second order Markov chains. Lastly, pitches were generated using a joint probabilistic distribution based on chords and the previous note. This system integrated the entire song writing process, from lyric generation to melody and accompaniment.

Another complete automated song writing system is the work by Scirea et al. [38]. Their system SMUG wrote songs from academic papers relying on the evolution of Markov chains. It used a data corpus of popular songs from mixed-genres, and integrated several rules [41].

Monteith et al. [39] studied the generation of melodic accompaniments. Along with a data corpus, their system works by generating hundreds of corpus-driven random options which are guided by a few rules, and then makes a choice from them based on an evaluation criterion that incorporates some musical knowledge [41].

Nichols [40] experimented on lyric-based rhythm suggestion where he considers the set of all accompanying rhythms and defines a fully rule-based function to evaluate them, via considerations such as rare word emphasis through strong beats. He also suggests as future work, to solve the rhythm generation process through machine learning, which is the approach followed by ALYSIA. ALYSIA did not encode any musical rules but used Random forests instead of Markov chains.

In contrast to our objective, in the work of Oliveira [42], the pipeline has been inverted, and lyrics were generated based on the rhythm of the provided melody.

In summary, the Literature Review pointed out the wide range of different methodologies that had been employed successfully in the domain of musical composition and also the current progress of the same in musical song writing. Inspired by the work of Doug [33], in our research we are looking at the possibility of using LSTM RNNs for musical song writing.

III. Methodology

The following are the main keys aspects of the methodology that was followed in our research work:

- Corpus Construction
  The proposed system is an entirely data-driven system. Therefore, it was crucial to construct a data corpus with essential features required by the neural network. Hence, the aim was to construct a data corpus that will allow the model to learn the mechanics and technicalities behind the structure of songs and their composition.
- Model Building and Evaluation
  The machine learning model had to be constructed and trained with the constructed data corpus, and then some sort of evaluation mechanism had to be defined, to evaluate the output of the model.
- Implementation of an Inference Tool
  The developed trained model had to be easily accessible to be used for inference, and hence, a web-based tool was developed.

Figure 1 summarizes the high-level architecture design for the proposed solution.

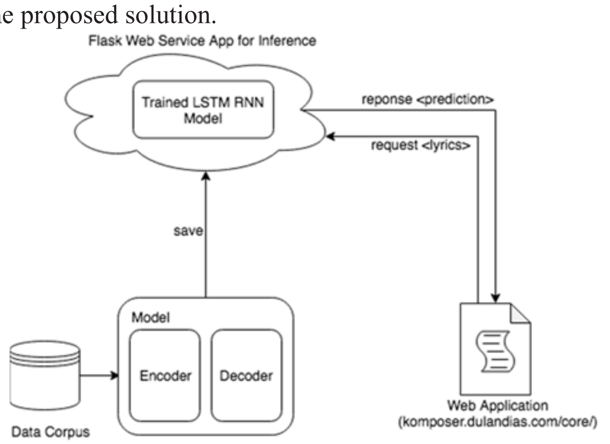

Figure 1. High-Level Framework of the Proposed Solution

A. Corpus Construction

Construction of the data corpus was the first challenging task that had to be undertaken. It was quite a challenging task due to the following reasons:

- Choosing a format to store the data (in audio or textual format) in such a way that we can maximize the features that can be stored while minimizing the processing required to extract features

- Finding a proper collection of musical records with lyrics and melodies accompanied, that can be used to construct the data corpus with consistency

Under Corpus Construction, the following steps were followed:

1. Choosing of a Data Storing Format

The first decision that had to be made was what format should be used to store the data, whether it should be in audio format or whether it could be done via a textual format.

When the word "music" first comes into our mind, what we think of is that we need to store the data in audio format in order to be used for processing. Therefore, in the case of audio format, we will have to find melodies (which will be in .mp3 or .MIDI format) as well as their accompanying lyrics (which will be in textual format, separately) to be stored in the data corpus.

In our other option, which is storing both of the melody and the accompanying lyrics in textual format, we had to search for existing standards which are used for other music related purposes. We were able to find two such standards which are used in storing musical melody sheets in textual notation. MusicXML [43] and abc-notation [44] were the two such standards. However, while abc-notation is capable of storing both the melody and lyrics in the same file, MusicXML was only able to store the melody in textual format.

Based on the comparison in Table I, it was decided to choose the textual format in constructing our data corpus. abc-notation was chosen over MusicXML due to the fact that it can store both the lyrics and the melody in a single file.

TABLE I: TABULAR COMPARISON FOR AUDIO VS TEXTUAL DATA STORING FORMATS FOR DATA CORPUS

Considered Factor	**Format**	
	Audio	**Textual**
Number of features that can be stored	High	High
Amount of processing time to extract features	Slow	Fast
Amount of processing time for analysis	Slow	Fast
Required storage capacity	High	Low
Availability of existing sources for data extraction	High	Normal
Store melody and/ or lyrics	Melody only	Melody and Lyrics

2. Creating a Consistent Data Corpus with Existing Records

In order to create a consistent data corpus with existing records that are spread over the world-wide web, we had to ensure the following:

- Availability of sufficient amount of records in abc-notation
- Records contain both the melody as well as the lyrics
- Amount of and the types of feature information stored in all the records are identical

The decision of choosing of abc-notation as the standard to store musical melodies in our data corpus would become an utmost failure if we were not able to find sufficient existing data records that can be transformed and used for our purpose. Making it most challenging, there were only a scarce amount of such sources where they had musical melodies with lyrics stored in abc-notation. Hence, another decision had to be made to construct the data corpus with the existing records but by not measuring the depth of the data corpus with the amounts of songs listed in the data corpus, but by the number of lines of melody and lyrical accompaniments that could be extracted with the amount of songs we had in abc-notation. In choosing of the selected songs in abc-notation, we had to filter most existing records because most of them did not have both the melody and lyrics stored together, while the rest were filtered out to make sure the data corpus contains songs of similar genres of music.

C. Model Building and Evaluation

Our objective is to attempt to solve the research problem using a Long Short-Term Memory (LSTM) Recurrent Neural Network implementation. Choosing a suitable development framework for the neural network implementation for training and inference was the first challenging task in this phase. However, due to the quick learning curve and support for LSTM RNNs, Keras with TensorFlow was chosen over other considered options such as Theano and Caffe.

1. Model Building and Training

We implemented a character-level sequence-to-sequence model, where the input character was processed character-by-character and the output was also generated character-by-character.

The following steps were followed in the training process:

(i) Turn the lyrical lines into 3 arrays:
- *encoder_input_data* is a 3D array of shape (number of pairs, maximum lyrical input length, number of English characters) containing a one-hot vectorization of the English lyrical lines.
- *decoder_input_data* is a 3D array of shape (number of pairs, maximum melody line length, number of melody characters) containing a one-hot vectorization of the melody lines in abc-notation.
- *decoder_target_data* is the same as *decoder_input_data* but offset by one time step.

(ii) Train a basic LSTM-based model to predict *decoder_target_data* given *encoder_input_data* and *decoder_input_data*, using teacher-forcing.

(iii) Decode lyrics to check that the model is working.

It is important to note that we did not split our data corpus into two portions for training and testing purposes. This is due to the fact that the generated output of our model is a melody that is supposed to accompany the provided lyrics. However, given a lyrical input, it could have multiple matching musical melodies to accompany it. Therefore, if the generated output is not identical to the expected output, we cannot necessarily say that the generated output was invalid. This issue makes it challenging for us to validate our trained model. This is addressed in the next section under Model Evaluation and Validation.

Figure 2 shows a high-level summary of the above training process, while Figure 3 shows a snapshot summary of our training model.

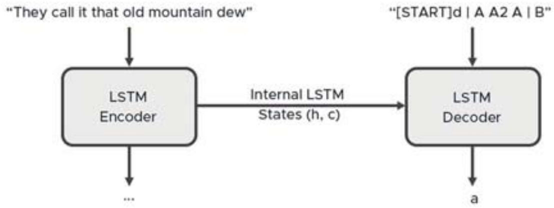

Figure 2. High-Level Summary of the Training Process

Layer (type)	Output Shape	Param #	Connected to
input_9 (InputLayer)	(None, None, 62)	0	
input_10 (InputLayer)	(None, None, 40)	0	
lstm_5 (LSTM)	[(None, 256), (None,	326656	input_9[0][0]
lstm_6 (LSTM)	[(None, None, 256),	304128	input_10[0][0] lstm_5[0][1] lstm_5[0][2]
dense_3 (Dense)	(None, None, 40)	10280	lstm_6[0][0]

```
Total params: 641,064
Trainable params: 641,064
Non-trainable params: 0
None
```

Figure 3. Snapshot Summary of the Training Model

For inferring, we will repeatedly:

(i) Encode the input line and retrieve the initial decoder state.

(ii) Run one step of the decoder with this initial state and a "start of sequence" token as target. The output will be the next target character.

(iii) Append the target character predicted and repeat.

Figure 4 shows a high-level summary of the above inference process, while Figure 5 and Figure 6 show snapshot summaries of our models used in the inferring process.

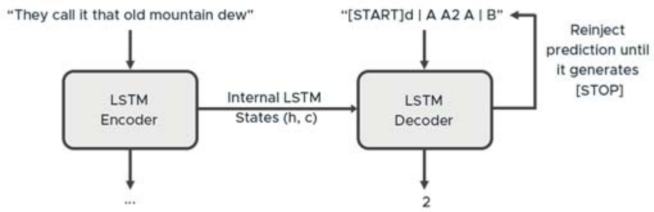

Figure 4. High-Level Summary of the Inference Process

Layer (type)	Output Shape	Param #
input_9 (InputLayer)	(None, None, 62)	0
lstm_5 (LSTM)	[(None, 256), (None, 256)	326656

```
Total params: 326,656
Trainable params: 326,656
Non-trainable params: 0
None
```

Figure 5. Snapshot Summary of the Encoder Model

Layer (type)	Output Shape	Param #	Connected to
input_10 (InputLayer)	(None, None, 40)	0	
input_11 (InputLayer)	(None, 256)	0	
input_12 (InputLayer)	(None, 256)	0	
lstm_6 (LSTM)	[(None, None, 256),	304128	input_10[0][0] input_11[0][0] input_12[0][0]
dense_3 (Dense)	(None, None, 40)	10280	lstm_6[1][0]

```
Total params: 314,408
Trainable params: 314,408
Non-trainable params: 0
None
```

Figure 6. Snapshot Summary of the Decoder Model

2. *Model Evaluation and Validation*

Model Evaluation and Validation was carried out in two methods:

(i) Through a Consistency Evaluation Algorithm

As was mentioned earlier, we did not split our data corpus into two portions for training and testing purposes. In this scenario, we cannot perform any traditional mathematical operations to calculate the accuracy of our trained model, and neither doing a sequence-to-sequence matching of the generated output vs the expected output will be appropriate. Therefore, a new algorithm had to be introduced in order to evaluate and validate the trained model.

The key idea behind this algorithm is the fact that since the model was trained with abc-notation, the generated output should follow the rules of the abc-notation standard, and thereby, the model should learn some of the basic rules of music. The rule that we will be specifically looking at is how well the trained model has learned the fact that the generated output should contain the same number of notes per each sector of the melody. That is, if it is a 4-by-4 melody, it should have 4 notes in each sector.

Proposed Consistency Evaluation Algorithm follows the following steps in determining the consistency of the generated output:

- Count the number of notes that exist in each sector.
- Group and count the number of sectors with same number of notes in them.
- Find the most frequently appearing group, and calculate the consistency using the equation below:

$$c = \frac{count\ of\ sectors\ in\ most\ frequent\ group}{total\ number\ of\ sectors}$$

(ii) User Feedback Collection via Inference Tool

Since music is something that cannot be evaluated by a machine or any algorithm, feedback was collected from users via the web-based inference tool (available at http://sci.sjp.ac.lk/composer/) and analysed in order to understand the accuracy and validity of the generated outputs. Feedback was collected from both amateurs as well as professionals in the music industry.

D. *Implementation of the Inference Tool*

The built and trained machine learning model had to be accessible in such a way that we can freely use it for inferences purposes. Therefore, the trained model was saved and hosted on a cloud server, where a Flask App serves as an RESTful API accepting JSON requests to which predictions are inferred from the model and a JSON response is returned back.

IV. Results and Discussion

As was discussed in previous sections as well, the main method that is used for evaluation purposes is the feedback obtained from users via the inference tool. However, in addition to this a novel algorithm was developed and used in order to mathematically measure the consistency of the generated output which is directly proportional to the accuracy of the model.

It was crucial to determine the parameters for the training process that will yield the best output. Figure 7 shows a summary on how the model loss varies with the number of epochs up to 1000.

One of the most challenging tasks in our research work was overcoming the problem that the model cannot be validated using any traditional methods (such as accuracy metrics for the loss function), since the generated output is a musical melody and a given lyric could have multiple interpretations for a matching melody. The following graph in Figure 8 shows that the traditional validation loss measurement cannot be used to justify the results of our model.

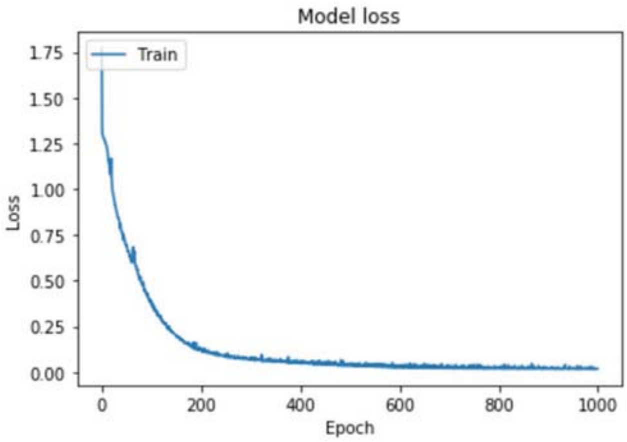

Figure 7. Model Loss vs Number of Epochs

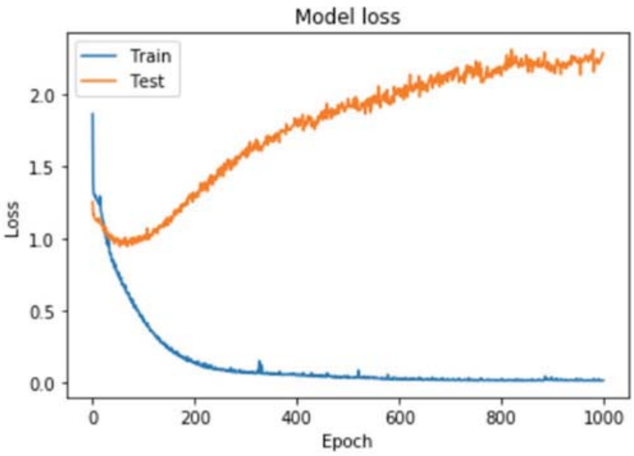

Figure 8. Model Training Loss and Testing Loss vs Number of Epochs

A. *Results based on User Feedback*

The web-based inference tool was shared between mainly two identified target groups, amateurs and professionals in music, and collected feedback. In this scenario, the user will give feedback to different outputs that are generated by Komposer. In addition to this, user feedback was collected by giving them to listen to a given melody generated by Komposer (results summarized in Table II).

TABLE II: USER FEEDBACK SUMMARY FOR A GIVEN MELODY

Feedback	**Frequency**
Pleasant to listen to there are well developed sub-structures of notes	22
Somewhat pleasant there are some good sub-structures of notes	3
Not pleasant to listen to there is no structure at all	0

Table III shows the summary of the user feedback that was received through the Inference Tool.

TABLE III: USER FEEDBACK SUMMARY FOR A GIVEN MELODY

Feedback for "How well did Komposer compose the melody to your lyrics"	**Response**	
	Frequency	**Percentage (%)**
Very Good	28	41.18
Good	37	54.42
Okay	3	4.4
Bad	0	0
Very Bad	0	0

The summary of feedback in Table III could be broken down into two segments based on whether the feedback provider was an amateur or a professional as shown in Figures 9 and 10.

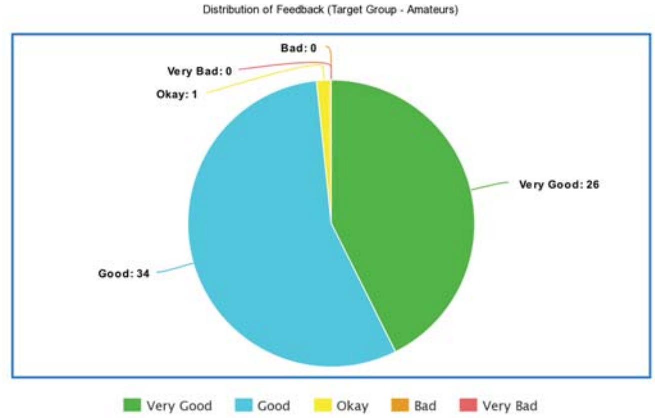

Figure 9. Pie Chart showing Distribution of Feedback from Amateurs

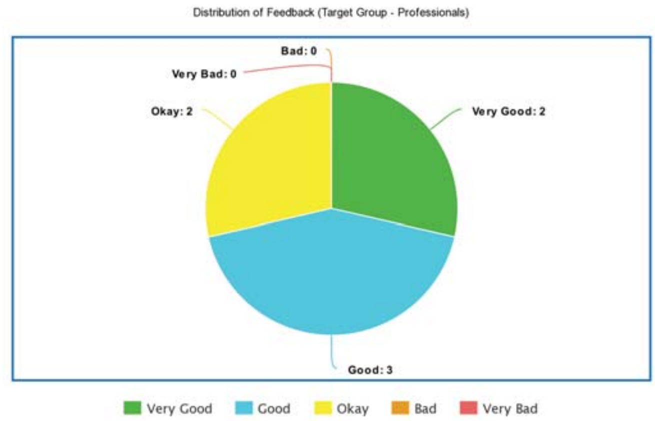

Figure 10. Pie Chart showing Distribution of Feedback from Professionals

B. *Results based on Consistency Evaluation Algorithm*

The Consistency Evaluation Algorithm gives a measure on how well the trained model has learned the basic rules of music and abc-notation, by evaluating the generated output. Let's take the generated output for a selected lyrical piece and compare the consistency values of the generated outputs when inferred from a model trained with 100 epochs and 1000 epochs. Table IV summarises the results obtained.

TABLE IV: SUMMARY OF RESULTS OBTAINED BY THE CONSISTENCY EVALUATION ALGORITHM

Number of Epochs Model was Trained On	Consistency (%)
100	75
1000	81.82

Therefore, it can be concluded that when the number of epochs a model is trained on is increased, it leads to both increment of the consistency as well as it brings more quality and structure to the generated outputs.

However, even at higher epoch numbers the generated output sometimes gave out tunes which are completely not in-line with the provided lyrics. This was identified to be caused due to the limitations from the depth of the data corpus.

V. Conclusion

The excellent results that were obtained and discussed in the previous section are key to the fact that we can legitimately call that the research objectives have been successfully met.

In summary, the following research objectives were accomplished with success:

- Different approaches and technologies currently employed in real world applications and in research level experiments were studied.
- New machine learning models and algorithms that have been applied in other subject domains based on their success were explored.
- A novel method was introduced to generate musical melodies based on provided lyrical inputs using LSTM RNNs.
- A web-based easily accessible inference tool was developed.
- A novel method was proposed and implemented to measure and validate the accuracy of the generated outputs via a consistency evaluation.

Besides achieving the targeted research Objectives, another very valuable deliverable was also created in the process of our research work, which is the construction of the data corpus. This data corpus is highly valuable for future research work in related domains, and we intend to make it publicly available for research students and academia.

As future work, the following aspects are some directions to which this project could be taken forward towards:

- Construct data corpora based on different genres of songs.
- Evaluate variations in consistency for different parameters of the LSTM RNNs.
- Implement a tool to generate and play the generated outputs with different musical instruments.
- Build a commercial tool that will assist musical composers to easily compose melodies for lyrics.

References

[1] Wikipedia. Music. URL https://en.wikipedia.org/wiki/Music.
[2] M. Ackerman and D. Loker. Algorithmic songwriting with alysia.arXiv:1612.01058 [cs], 2016. URL http://dx.doi.org/10.1002/andp.19053221004.
[3] Bruce L. Jacob. Composing with genetic algorithms. Proceedings of the International Computer Music Conference, 1995.
[4] P. Westergaard, L. Hiller and L. Isaacson, "Experimental Music. Composition with an Electronic Computer", Journal of Music Theory, vol. 3, no. 2, p. 302, 1959. Available: 10.2307/842857.
[5] Cope. Panel discussion. 1993.
[6] Cope, David. Experiments in musical intelligence (EMI): Non-linear linguistic-based composition. Journal of New Music Research. 18.117-139. 10.1080/09298218908570541. 1989.
[7] H. Noguchi. Musical game in c k.516f, 1997. URL http://www.asahinet.or.jp/ ~rb5h-ngc/e/k516f.html.
[8] M. Steedman. The blues and the abstract truth: Music and mental models. Mental Models in Cognitive Science, 1996.
[9] P. N. Johnson-Laird. Jazz improvisation: A theory at the computational level. Representing Musical Structure, 1991.
[10] C. Ames and M. Domino. Cybernetic composer: An overview. Understanding Music with AI, page 186205, 1992.
[11] I. Xenakis. Formalized Music. Indiana University Press, 1971.
[12] J. Pressing. Nonlinear maps as generators of musical pitch. Computer Music Journal, page 3546, 1988.
[13] M. Herman. Deterministic chaos, iterative models, dynamical systems and their application in algorithmic composition. 1993.
[14] J. Harley. Algorithms adapted from chaos theory. 1994.
[15] M. Gogins. Iterated functions systems music. Computer Music Journal, 15(1):3546, 1991.
[16] D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. Journal of New Music Research, 24:51 73, 1995.
[17] K. Ebcioglu. An expert system for harmonizing four-part chorales. Computer Music Journal, 12(3):4351, 1988.
[18] C. P. Tsang and M. Aitken. Harmonizing music as a discipline of constraint logic programming. 1991.
[19] F. Pachet and P. Roy. Formulating constraint satisfaction problems on part-whole relations: the case of automatic harmonization, 1998.
[20] G. Ramalho and J. G. Ganascia. Simulating creativity in jazz performance. National Conference on Artificial Intelligence, 1:108113, 1994.
[21] D. Zimermann. Modelling musical structures. aims, limitations and the artists involvement, 1998.
[22] T. Stapleford J. Robertson, A. de Quincey and G. Wiggins. Real-time music generation for a virtual environment, 1998.
[23] P. M. Todd and G. Loy. Music and Connectionism. MIT Press, 1991.
[24] M. Leman. Artificial neural networks in music research. Computer Representations and Models in Music, page 265301, 1992.
[25] N. Grith and P. M. Todd. Musical Networks. MIT Press, 1997.
[26] P. M. Todd. A connectionist approach to algorithmic composition. Computer Music Journal, 13(4):2743, 1989.
[27] M. C. Mozer. Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing, Connection Science. 1994.
[28] M. I. Bellgard and C. P. Tsang. Harmonizing Music the Boltzmann Way, Connection Science. 1994.
[29] P. Toiviainen. Modeling the target-note technique of bebop-style jazz improvisation: An artificial neural network approach, Music Perception. 1995.
[30] D. Hornel and P. Degenhardt. A neural organist improvising baroquestyle melodic variations. Proceedings of the International Computer Music Conference, 1997.
[31] D. Hornel. Melonet i: Neural nets for inventing baroque style chorale variations. Advances in Neural Information Processing 10 (NIPS 10), 1997.
[32] A. F. Melo. A connectionist model of tension in chord progressions, 1998.
[33] D. Eck and J. Schmidhuber. Finding temporal structure in music: Blues improvisation with lstm recurrent networks. 2002.
[34] J. H. Holland. Adaptation in natural and artificial systems. 1975.
[35] David E. Goldberg. Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989. ISBN 0201157675.
[36] M. Gutknecht. The postmodern mind: hybrid models of cognition. Connection Science. 1992.
[37] Hannu Toivonen Jukka M Toivanen and Alessandro Valitutti. Automatical composition of lyrical songs. In The Fourth International Conference on Computational Creativity, 2013.
[38] Noor Shaker Marco Scirea, Gabriella AB Barros and Julian Togelius. Smug: Scientific music generator. In Proceedings of the Sixth International Conference on Computational Creativity June, page 204, 2015.
[39] Tony Martinez Kristine Monteith and Dan Ventura. Automatic generation of melodic accompaniments for lyrics. In Proceedings of the International Conference on Computational Creativity, page 8794, 2012.
[40] Eric Nichols. Lyric-based rhythm suggestion. Ann Arbor, MI: Michigan Publishing, University of Michigan Library, 2009.
[41] M. Ackerman and D. Loker. Algorithmic songwriting with alysia. Journal of Artificial General Intelligence, arXiv:1612.01058, 2016.
[42] Hugo Gonc alo Oliveira. Tra-la-lyrics 2.0: Automatic generation of song lyrics on a semantic domain. Journal of Artificial General Intelligence, 6(1):87110, 2015.
[43] Michael Good. Musicxml for notation and analysis, 2001. URL https://www.musicxml.com/publications/makemusic-recordare/notation-and-analysis/.
[44] The abc music standard 2.1, 2011. URL http://abcnotation.com/wiki/abc: standard:v2.1.