LSTM based Music Generation with Dataset Preprocessing and Reconstruction Techniques

Sarthak Agarwal
Department of Computer Engineering
Netaji Subhas Institute of Technology
Dwarka, New Delhi-110078, India
sarthak18696@gmail.com

Vaibhav Saxena
Department of Computer Engineering
Netaji Subhas Institute of Technology
Dwarka, New Delhi-110078, India
vbhvsxn@gmail.com

Vaibhav Singal
Department of Computer Engineering
Netaji Subhas Institute of Technology
Dwarka, New Delhi-110078, India
vaibhav.singal1996@gmail.com

Swati Aggarwal*
Department of Computer Engineering
Netaji Subhas Institute of Technology
Dwarka, New Delhi-110078, India
swati1178@gmail.com

***Abstract*—Numerous approaches have been used by researchers for the purpose of music generation. Recurrent neural networks (RNNs) and Long short term memory (LSTM) networks are able to effectively model sequential data. LSTM networks have been extensively used to produce sheet music, character by character. These LSTM models, however, require a lot of time to train to be able to produce pleasant and syntactically correct sheet music. We introduce some effective dataset preprocessing and reconstruction techniques which facilitate the generation of syntactically correct sheet music, while reducing the training time. The quality of music generated is qualitatively measured by peers. The proposed model employing the dataset preprocessing and reconstruction techniques is compared with another model possessing no such techniques in a subjective manner.**

I. INTRODUCTION

Music composition is a process which requires skills and talent. Not everyone can compose music. But with the onset of innovative deep learning techniques, anyone could now generate novel and good sounding music.

Recurrent neural networks (RNNs) have become increasingly popular for sequence modeling. They can be applied to various domains like speech, text, video, or any kind of sequential data. However, recurrent neural networks possess short memories and are not able to learn long term dependencies. Long short term memory (LSTM) [1] networks have emerged as a solution to this problem. LSTMs are extensively applied to many deep learning problems like text summarization, text generation and many natural language modeling problems.

Long Short Term Memory (LSTM) networks can be used for the purpose of music generation as discussed in section II. Music is represented in ABC format [2], which is a textual form of music. The LSTM model uses character level modeling to generate music in the ABC format, by predicting musical notes character by character. The problem with the existing works is that the LSTM models take a long time to produce syntactically correct ABC nototion of pleasant sounding music.

*Corresponding Author

In this work, we present some novel dataset preprocessing and reconstruction techniques which facilitate fast training, and generation of musical notes possessing correct ABC notation syntax. The proposed LSTM model works on the **Pre**processed dataset, and then proper **Recon**struction techniques are used to generate sheet music. Hence, the model is called **PRECON-LSTM**. We show the advantages of these techniques by comparing PRECON-LSTM with a model employing no such dataset preprocessing and reconstruction techniques.

In the following sections, we discuss the related work in section II, followed by an in depth explanation of the proposed model and techniques in section III. Finally, we discuss the experiments and results in section IV, and concluded in section V.

II. RELATED WORK

Different approaches have been adopted by researchers over the years for the purpose of music generation. They have tried to use different mathematical models like Markov chains, knowledge based systems and multiple deep learning approaches to compose novel and good sounding music.

Xenakis [3] used Markov chains to artificially compose music. Markov chains were used to predict the most likely musical note, depending on the previous musical notes in the Markov chain. While Markovian models are good options for real time applications since they are relatively less complex, these models fail to produce novel music, and can only produce musical sequences which are already existent in the original data.

Knowledge based methods like genetic algorithms have proven to work very well for the purpose of music composition. Researchers (Horner and Goldberg [4]) have used these algo-

978-1-5386-9276-9/18/$31.00 ©2018 IEEE

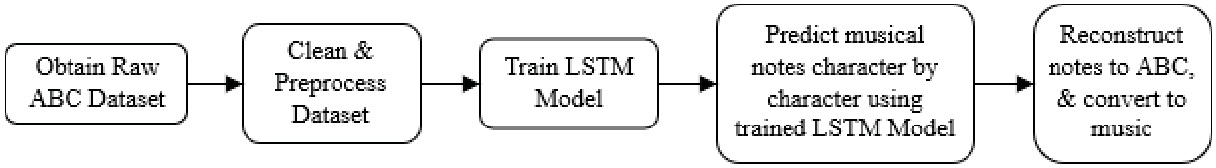

Fig. 1. Flowchart for PRECON-LSTM

rithms for composition since they are efficient search methods, and have the ability to offer multiple solutions to a particular problem, proving to be perfect for music composition which demands creative results. The problem with using genetic algorithms is that they heavily depend on the amount of knowledge the system possesses, making the process inefficient. Also, these algorithms try to minimize the search space by using simplistic representations of music, which leads to loss of musical quality.

Use of artificial neural networks have become extremely popular for music generation, by being able to predict musical notes. The main advantage of using neural networks is that they learn the musical structure, and do not require a human expert to impart musical knowledge and analysis. Recurrent neural networks (RNNs) and Long short term memory (LSTMs) networks are being used because of their capability to keep track of the musical notes generated in the past, and accordingly predict new musical notes.

Todd [5] and Mozer [6] used a recurrent neural network (RNN) for music composition. The results produced by them were fair, but sounded repetitive and monotonous. Due to the inability of RNNs to store long term dependencies, researchers have started using LSTMs since they can easily store context in a given sequence of data.

Eck et al. [7] use a standard LSTM model to learn fixed and simple chord structure. They represent music in the standard MIDI format. The issue with such a representation is that the standard MIDI format is continuous in nature, and may not be able to get converted to sheet music and be transcribed properly. Karpathy [8] developed an LSTM model, called CharRNN, which takes as input, a large corpus of text for training, and predicts the output character by character. Johnston [9] and Huang et al. [10] use the model introduced by Karpathy for the purpose of music generation. In these works, music is represented in the ABC notation, which is a textual representation of music. These works predict the following musical notes character by character. Sturm et al. [11] use a similar model, but perform very basic preprocessing to the input ABC dataset.

The issue with the above mentioned works is that they are prone to produce ABC notation with syntactical errors, which may lead to an unpleasant music synthesis, or even no music synthesis. In this work, we present an LSTM based music generating model, called PRECON-LSTM, which employs effective dataset preprocessing and reconstruction techniques, which ensure the production of syntactically correct music in ABC format, while producing pleasant sounding music.

III. MODELING AND GENERATION

A Long Short Term Memory (LSTM) model, called PRECON-LSTM, is used for the purpose of music generation. The entire process of music generation is explained in depth in the further subsections, with the help of an example.

Dataset preprocessing and reconstruction techniques have been employed by the model, which reduce the size of the training dataset by removing unimportant information, thus reducing the training time. Also, since the unnecessary information is removed, the model's only focus is to learn the textual musical notes. Hence, for the same number of epochs, the model is able to make better predictions, as discussed in section IV-B.

First, the dataset used for training PRECON-LSTM is described, followed by an explanation of the adopted dataset preprocessing techniques. Next, the LSTM model and its process of training is described in detail. Finally, the process of musical notes prediction, along with the employed reconstruction techniques are explained. The entire process is illustrated in the form of a flowchart in fig. 1.

A. Dataset Description

For the purpose of this work, the Nottingham Folk Music dataset [12], a standard dataset representing music in ABC notation, was used. It consists of a collection of more than 1000 folk tunes (using jigs, hornpipes, reels etc.) in textual format using the ABC notation. The ABC notation [2] is an accepted standard developed by Walshaw, which represents music using a specified syntax written in plain text format. An example of a musical tune in its ABC format is shown in fig. 2.

Every musical tune in the ABC format comprises of two parts: the header and the musical notes. The first 7 lines in fig 2 represent the header part, which gives the information about the tune reference number (X), title (T), comments (%), composer (S), meter (M), base chord (K) and note length (L). It represents all the background information other than the harmonic tunes which is related to the music. The M field denotes the meter of the song, which refers to the regularly recurring patterns and accents such as bars and beats. The value left to "/" represents number of beats per measure. The K field denotes the base chord, and can take a value from A to G. The L field in header describes the note length. 1/4 means that each letter in the note section is of a quarter-note length.

The musical notes section of the ABC notation contains the information of the notes of the tune. Letters A to G represent each chord. Uppercase letters represent the higher octave

```
X: 1
T:Blaydon Races
% Nottingham Music Database
S:Kevin Briggs, via EF
M:6/8
K:D
L:1/4
A>B c<d e#f|"D"d2d d2d|"D"d2d d2d|"A7"e2e e2e|\
"D"f3 "D7"d3|"G"ggg g2g|"D"f2f f2f|
"E7"e2f e2d|"A7"c3 A3|"D"d2d d2d|"D"d2d d2d|\
"A7"e2e e2e|"D"f3 "D7"d3|"G"ggg g2g|
"D"f2g a2a|"A7"a2g f2e|"D"d2e f2g|"D"a3 "A7/e"a2f|\
"D/f+"d3 "G"d2d|"E7/g+"e2e "A7"e2e|
"D"f3 "D7/c"d3|"G/b"g2g "A7/c+"g2g|"D"f2f "D/f+"f2f|\
"G"e2f "E7/g+"e2d|"A"c3 "A7/g"A3|"D/f+"d2d "G"d2d|\
"D/f+"d3 "G"d2d|
"E7/g+"e2e "A7"e2e|"D"f3 "D7/c"d3|\
"G/b"g2g "A7/c+"g2g|"D"f2g "B7/f+"a2a|\
"E"a2g "A7"f2e|"D"d3 d2||
```

Fig. 2. Example of music in ABC notation

notes and lowercase letters the notes in the lower octave.
The dataset comprises of musical tunes from the Nottingham music database, which is further processed using certain techniques described in the following sections.

B. Dataset Preprocessing and Cleaning

B.1 First step of Preprocessing: The first step of cleaning the text is removing the headers and comments from the text, since only the musical notes are important for the generation of music. The example in fig. 2, after removing the headers and the comments, is reduced to the form as shown in fig. 3.

```
A>B c<d e#f|"D"d2d d2d|"D"d2d d2d|"A7"e2e e2e|\
"D"f3 "D7"d3|"G"ggg g2g|"D"f2f f2f|
"E7"e2f e2d|"A7"c3 A3|"D"d2d d2d|"D"d2d d2d|\
"A7"e2e e2e|"D"f3 "D7"d3|"G"ggg g2g|
"D"f2g a2a|"A7"a2g f2e|"D"d2e f2g|"D"a3 "A7/e"a2f|\
"D/f+"d3 "G"d2d|"E7/g+"e2e "A7"e2e|
"D"f3 "D7/c"d3|"G/b"g2g "A7/c+"g2g|"D"f2f "D/f+"f2f|\
"G"e2f "E7/g+"e2d|"A"c3 "A7/g"A3|"D/f+"d2d "G"d2d|\
"D/f+"d3 "G"d2d|
"E7/g+"e2e "A7"e2e|"D"f3 "D7/c"d3|\
"G/b"g2g "A7/c+"g2g|"D"f2g "B7/f+"a2a|\
"E"a2g "A7"f2e|"D"d3 d2||
```

Fig. 3. First step of Preprocessing

B.2 Second step of Preprocessing: The '|' symbol in the musical notes acts as a delimiter, and divides the musical paragraph into groups of notes called measures. Each measure is represented by a set of notes, where each note is followed by its coefficient. For example, '*e2d*' in the eighth measure of the example mentioned (fig. 2), means two '*e*' notes and one '*d*' note. According to the syntactical rules of ABC notation, the sum of the coefficients in one measure should be equal to number of beats per measure specified in the meter of the song. Since this piece of information is also not related to musical notes as well, it is removed. After this step, the text is reduced to a form shown in fig. 4.

```
A>B c<d e#f"D"d2d d2d"D"d2d d2d"A7"e2e e2e"
D"f3 "D7"d3"G"ggg g2g"D"f2f f2f
"E7"e2f e2d"A7"c3 A3"D"d2d d2d"D"d2d d2d
"A7"e2e e2e"D"f3 "D7"d3"G"ggg g2g
"D"f2g a2a"A7"a2g f2e"D"d2e f2g"D"a3 "A7/e"a2f
"D/f+"d3 "G"d2d"E7/g+"e2e "A7"e2e
"D"f3 "D7/c"d3"G/b"g2g "A7/c+"g2g"D"f2f "D/f+"f2f
"G"e2f "E7/g+"e2d"A"c3 "A7/g"A3"D/f+"d2d "G"d2d
"D/f+"d3 "G"d2d
"E7/g+"e2e "A7"e2e"D"f3 "D7/c"d3
"G/b"g2g "A7/c+"g2g"D"f2g "B7/f+"a2a\
"E"a2g "A7"f2e"D"d3 d2
```

Fig. 4. Second step of Preprocessing

```
A>Bc<de#f"D"d2dd2d"D"d2dd2d"A7"e2ee2e"
D"f3"D7"d3"G"gggg2g"D"f2ff2f
"E7"e2fe2d"A7"c3A3"D"d2dd2d"D"d2dd2d
"A7"e2ee2e"D"f3"D7"d3"G"gggg2g
"D"f2ga2a"A7"a2gf2e"D"d2ef2g"D"a3"A7/e"a2f
"D/f+"d3"G"d2d"E7/g+"e2e"A7"e2e
"D"f3"D7/c"d3"G/b"g2g"A7/c+"g2g"D"f2f"D/f+"f2f
"G"e2f"E7/g+"e2d"A"c3"A7/g"A3"D/f+"d2d"G"d2d
"D/f+"d3"G"d2d
"E7/g+"e2e"A7"e2e"D"f3"D7/c"d3
"G/b"g2g"A7/c+"g2g"D"f2g"B7/f+"a2a
"Em"a2g"A7"f2e"D"d3d2
```

Fig. 5. Third step of Preprocessing

B.3 Third step of Preprocessing: Various special characters were analyzed, and their relevance to the musical information of a tune were checked. Following are the observations made regarding the special characters occurring in the musical notes:

- [Whitespace, '\n', %] hold no importance.
- **[ ]** displays a chord, but **""** can also be used instead of them.
- **=** displays a natural note but omitting it does not have any effect.
- \ is used to continue the section to another line. Hence, it is removed and the two sentences are joined.
- [**! .** ∼] denote accents (!segno! , !trill! , staccato , turn). They are removed because *abc2midi* [13], a package used for the conversion of ABC notation to its music form, recognizes them but brings no change in the music generated.
- [{} **,** ()] denote tuplets, slurs and ornaments. They are not handled in this paper and remain a limitation of this work.

After implementing the above made observations in the ongoing example, the text is reduced to a form shown in fig. 5.

B.4 Fourth step of Preprocessing: According to the syntactical rules of the ABC notation, anything of the form x>y and x<y, where x and y are valid notes, can be represented by x3/2y/2 and x/2y3/2 respectively. Therefore, they are replaced. The new form of the text is shown in fig. 6.

```
A3/2B/2c/2d3/2e#f"D"d2dd2d"D"d2dd2d"A7"e2ee2e
"D"f3"D7"d3"G"gggg2g"D"f2ff2f
"E7"e2fe2d"A7"c3A3"D"d2dd2d"D"d2dd2d
"A7"e2ee2e"D"f3"D7"d3"G"gggg2g
"D"f2ga2a"A7"a2gf2e"D"d2ef2g"D"a3"A7/e"a2f
"D/f+"d3"G"d2d"E7/g+"e2e"A7"e2e
"D"f3"D7/c"d3"G/b"g2g"A7/c+"g2g"D"f2f"D/f+"f2f
"G"e2f"E7/g+"e2d"A"c3"A7/g"A3"D/f+"d2d"G"d2d
"D/f+"d3"G"d2d
"E7/g+"e2e"A7"e2e"D"f3"D7/c"d3
"G/b"g2g"A7/c+"g2g"D"f2g"B7/f+"a2a
"Em"a2g"A7"f2e"D"d3d2
```

Fig. 6. Fourth step of Preprocessing

B.5 Fifth step of Preprocessing: ABC notation supports two formats for representing a sharp note, ˆa and a#. Only the former is used, and therefore the latter was converted to the first one, as shown in fig. 7. The form of musical notes produced here is final, and is then used for the purpose of training PRECON-LSTM.

```
A3/2B/2c/2d3/2^ef"D"d2dd2d"D"d2dd2d"A7"e2ee2e
"D"f3"D7"d3"G"gggg2g"D"f2ff2f
"E7"e2fe2d"A7"c3A3"D"d2dd2d"D"d2dd2d
"A7"e2ee2e"D"f3"D7"d3"G"gggg2g
"D"f2ga2a"A7"a2gf2e"D"d2ef2g"D"a3"A7/e"a2f
"D/f+"d3"G"d2d"E7/g+"e2e"A7"e2e
"D"f3"D7/c"d3"G/b"g2g"A7/c+"g2g"D"f2f"D/f+"f2f
"G"e2f"E7/g+"e2d"A"c3"A7/g"A3"D/f+"d2d"G"d2d
"D/f+"d3"G"d2d
"E7/g+"e2e"A7"e2e"D"f3"D7/c"d3
"G/b"g2g"A7/c+"g2g"D"f2g"B7/f+"a2a
"Em"a2g"A7"f2e"D"d3d2
```

Fig. 7. Fifth step of Preprocessing

C. Model Description and Training

The musical notes from the ABC notation of a tune were extracted and processed to remove unnecessary symbols, as described in section III-B. An LSTM model, named PRECON-LSTM, is used here because of the prowess of an LSTM model to learn patterns and produce commendable results, and the need to store long term dependencies involved in the structure of musical notes.

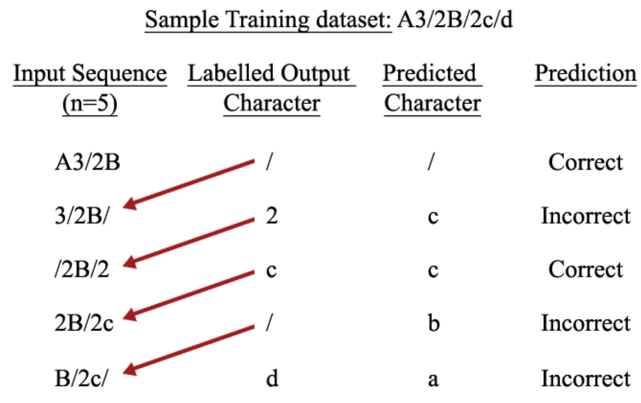

Fig. 8. LSTM training procedure

The idea is to give a set of musical characters represented in their textual form to the LSTM model as input. The LSTM model, in turn, should be able to predict the next musical note in the sequence. In order to do that, a sequence of '*n*' characters is taken as an input to the LSTM model in the form of a one hot encoded two dimensional vector, with the $(n+1)^{th}$ character as the one hot labelled output. SThe LSTM model, when given a sequence as input, predicts a character. This predicted character is compared with the corresponding labelled output character, and the error is backpropagated. A window of '*n*' characters translated over the entire text dataset generates different input sequences and compares the predicted character with the labelled output character, which helps in training the model. Fig. 8 illustrates the training procedure of the LSTM model.

PRECON-LSTM consists of an LSTM layer of 100 hidden neurons, followed by the softmax activation layer. The model uses Adagrad [14] as an optimizer, and is trained for 1500 epochs by minimizing the categorical cross-entropy loss function [15]. For the purpose of this work, on observing the results obtained with different values of '*n*' (the length of the input sequence), a value of 25 was chosen. A value of 25 helps to keep enough context in the character sequence, while not making the training process computationally inefficient. The training was done on a machine with Intel Core i5 4460K CPU and NVidia 750Ti GPU, which consists of 640 CUDA cores.

D. Musical Notes Prediction

A random sequence of characters of length '*n*' is obtained from the dataset file, and fed as input to the model. This random sequence acts as a seed for the model to start predicting the next musical notes. For this sequence, the $(n+1)^{th}$ character is predicted by the trained model. The predicted character, along with '*n-1*' characters of the previous sequence are fed to the model as the next input. This process is repeated recursively to obtain the predicted text of musical notes. Fig.

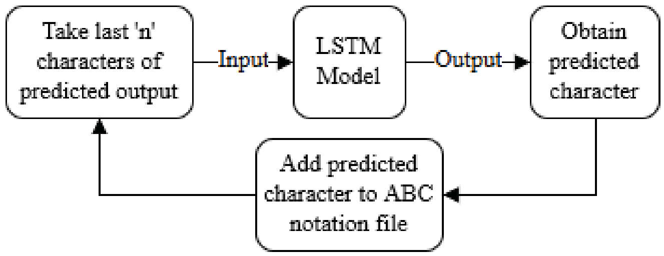

Fig. 9. Prediction Process Flowchart

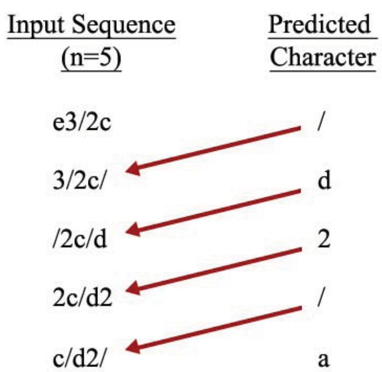

Fig. 10. Prediction Procedure

10 illustrates the prediction process when given a random sequence as input.

E. Reconstruction of Predicted Notes to Music Form

Suppose the predicted musical notes are:

e3/2c/2Ad2fA^ecADd2fA^ecA^ecABBcBEd2f
AecADdefAecADd2fDfafAedcBBcBEd2f
AecAacAAecADd2fAecAacABBcBEd2f
AecAacAAecADd2fDfafAedcBBcBEd2f

Fig. 11. Predicted Musical Notes

E.1 First step of Reconstruction: The headers removed in the first step of cleaning have to be added. The instrument choice, tempo, meter and key are provided by the user through a graphical user interface. Fig. 12 shows the predicted notes along with the header information according to the choices made by the user.

E.2 Second step of Reconstruction: Next, the following syntactical checks are performed on the predicted ABC notation file:

- Flat symbol ‘,’ should occur after a capital note, as it is required by the *abc2midi* package for converting ABC notation to its music form.
- Numbers, ‘/’, ‘+’, “ ’ ” should occur after valid notes.
- Consecutive numbers and special characters are not allowed.

X: 1
T: PREDICTION
M: *meter input from user*
K: *key input from user*
Q: *tempo input from user*
V: 1
%%*MIDI program instrument choice number from user*
e3/2c/2Ad2fA^ecADd2fA^ecA^ecABBcBEd2f
AecADdefAecADd2fDfafAedcBBcBEd2f
AecAacAAecADd2fAecAacABBcBEd2f
AecAacAAecADd2fDfafAedcBBcBEd2f

Fig. 12. First Step of Reconstruction

- After ‘_’ and ‘^’ there should be a valid note.
- ‘|’ and ‘-’ should be surrounded by valid notes.
- If a single ‘:’ is predicted in a measure, it is shifted to the end of the measure before ‘|’, and if two ‘:’ are predicted in a measure, the measure is ended used a ‘::’ instead of a ‘|’.

The ‘|’ character is placed when the sum of coefficients becomes equal to the number of beats per measure specified in the meter of the song (assuming meter= 6/8 in the ongoing example).

X: 1
T: PREDICTION
M: *meter input from user*
K: *key input from user*
Q: *tempo input from user*
V: 1
%%*MIDI program instrument choice number from user*
e3/2c/2Ad2f | A^ecaAD | d2fA^ec | A^ecABB | cBEd2f |
AecADd | efAecA | Dd2fDf | afAedc | BBcBEa | d2f
Aec | AacAAe | cADd2f | AecAac | ABBcBE | d2f
Aec | AacAAe | cADd2f | DfafAe | dcBBcB |

Fig. 13. Second Step of Reconstruction

After a valid ABC notation is reconstructed, it has to be converted to its musical form in order to generate actual music. The obtained ABC notation is converted to MIDI format using the *abc2midi* [13] package.

IV. RESULTS AND EVALUATION

The manner in which everyone perceives a particular kind of music is different. It is difficult to objectively evaluate a piece of music, as suggested by multiple works in this area of research [7], [16]. The quality of music can however, be subjectively evaluated, as shown in sub-section IV-A. Additionally, the effectiveness of the dataset preprocessing and reconstruction techniques employed by PRECON-LSTM have been evaluated in sub-section IV-B.

A. Subjective Evaluation of Music Generated

This work uses a method for the subjective evaluation of the generated music similar to the method used in work done by Huang et al [10].

To subjectively evaluate the quality of music produced, a survey is conducted among peers, in order to receive feedback on the quality of music produced. All subjects belong to an age group of 20-25 years, each possessing varied taste in music, with no formal musical training. In each trial, a subject is made to listen to 20 musical tunes, comprising of 10 original human composed musical tunes taken from the Nottingham Music Dataset itself, and 10 musical tunes generated by the proposed model, with the subject being unaware of which musical tune is real and which is fake. Each subject is asked to perform two tasks:

1) Rate the quality of the musical tune on a scale of 0-5, where 0 corresponds to an unpleasant musical tune, and 5 corresponds to a good sounding and pleasant musical tune.
2) Rate the realness of the musical tune on a scale of 0-5, where 0 corresponds to an artificially generated musical tune, and 5 corresponds to a human composed musical tune.

30 test subjects were used for the purpose of this work, and the results obtained are illustrated in table I.

TABLE I
MUSIC SURVEY RESULTS

Music Type \ Score	Mean Quality	Mean Realness
Artificially Generated	3.120	2.747
Human Composed	3.613	3.516

A mean quality score of 3.120 corresponding to the music generated by the proposed model, when compared to the score 3.613 obtained by human composed music, clearly indicates that the music produced is of comparable quality as that of a human composed musical tune. The model is able to generate pleasant, good sounding.

The music generated by the proposed model receives a mean realness score of 2.747 , while the human composed musical tunes receive a score of 3.516. This shows that the music generated by the proposed model is close to being indistinguishable to human composed music. However, the model shows traces of artificiality, and hence receives a less realness score than human composed music.

The quality scores and the realness scores achieved by the 10 musical tunes generated by the model are graphically illustrated in fig. 14 as well.

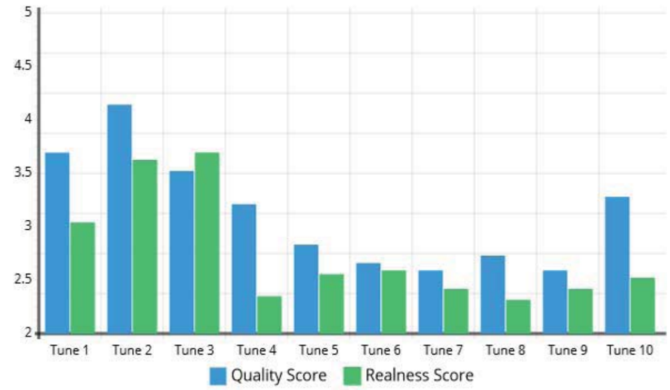

Fig. 14. Quality and Realness Scores of Generated Music

B. Evaluating the Dataset Preprocessing and Reconstruction Techniques

PRECON-LSTM employs powerful dataset preprocessing and reconstruction techniques. In order to exhibit the advantages of using such techniques, we compare the proposed model to another exactly same model which does not employ the adopted techniques.

Let the model which does not employ the adopted techniques be Model A. Model A is trained on unprocessed ABC files, and tries to learn the ABC syntax, as well as the musical notes pattern. PRECON-LSTM, on the other hand, is trained only on the musical notes, and its job is to learn musical notes only. Appropriate reconstruction techniques are used to convert the predicted musical notes to a valid ABC notation, as discussed in section III-E.

Both the models are compared at three different training instances, i.e. at 500 epochs, at 1000 epochs and at 1500 epochs. Both models are given an '*n*' length input seed text. Model A is given the first three headers of a total length '*n*' characters as the input seed text, while PRECON-LSTM is given some random '*n*' musical notes characters as the input seed text at all three stages.

1) *At 500 epochs*

From fig. 15, it can be seen that Model A has barely learned the structure of the ABC notation. The model has straightaway started to predict musical notes. The musical notes produced are syntactically wrong as well, and make no sense, since some of the characters predicted within a measure are not valid note characters, and the sum of coefficients in most of the measures is not equal to the number of beats per measure specified in the meter of the song. These invalid note characters are predicted since the model is trained on musical notes as well as different header statements which may contain invalid note characters. The produced characters do not make up a valid ABC notation, and hence can't be converted into its musical form. The model has, however, captured a few details, like the underlying patterns of musical notes.

Fig. 16 shows that PRECON-LSTM at 500 epochs is able to generate syntactically correct music because of the adopted reconstruction techniques. The LSTM model is able to predict musical notes correctly. However, the notes produced are a bit repetitive in nature, which indicates that the model has

```
X:1
T:PREDICTION
M:6/8
d|d2b ba|"G"f|a2g g2|"G"g|QGryLrke ry!GGrllG|
"C"rlrklkyr6|L3L%eG)pe|appppkpYpGpllkl2)G%e|
)JeJGeapmapYpkp1|lGa.a..JkaOmaOJOG|apal4|k?l2)
Ge)JeG|)l6|L6|Gapar2lkr1|rGaOaL2LkL1|LGapar2lka|
Oma4Ga|pal5|k?l2)Ge)Jel1|L6|Gapar2lkrLlG|a.aJ2JkaOma|
O2JGapal1|OkO2lGapal1|lkaOmaL2Ga|pal2lkaOmaL1|
rGapal3|kl2)Ge)Jel1|Gapal2rkrLlG|a.aJ2JkJ2|lGaOa4|
OkOJ4Ga|pal2lkaOmaLrLG|apal4|kl2)Ge)Jel1|L6|
Gapar2lkrLlG|a.aJ2JkaOma|O2JGapal1|OkO2lGapal1|
lkaOmaL2Ga|pal2lkaOmaL1|rGapal3|kl2)Ge)Jel1|
Gapal2rkrLlG|a.aJ2JkJ2|lGaOa4|OkOJ4Ga|pal2lkaOma
LrLG|apal4|
```

Invalid Note Character   Not consistent with no. of beats per measure specified in meter

Fig. 15. Model A at 500 epochs

```
X:1
T:PREDICTION
M:6/8
K:G
Q:1/2=60
V:1
%%MIDI program 0
EGEEBd|"G"BAFAdc|"D"edbeec|"C"eb5|ze/2b4z/2|
ebEdzb1|zEGf3|EzEd3|E6|Edd3E|EzEd3|E6|F+dzEzd1|
bE,bEzd|EzEd3|zE+zbdb|ebdeE,2|zEzzG+G|dG+zCzg|
GggGC2|g3dG+z|EzEd3|zC3G+G|+b3z3|zE+bde2|
bE+bdeb|E+b5|A'3bbd|bG+ceG+2|zdbdde::bE+bE+G+z1|
zd3z2|zbbEbE|+b6|A'3d3|zz3z2|G+G5|dA+G+zdz|
bdEbzz|dbG+Edz|EzEd3|"E"d3A'3|zzE+bEb|
```

Repetitive musical notes

Fig. 16. PRECON-LSTM at 500 epochs

```
X:1
T:PREDICTION
M:6/8
K:GE/
EEC|DEF E1|"A"D|vffxexGe2|"D"CBxF+MxvbMf3|
G!3vM3||Mf2Xx[fXEf|vfEC5bbCx xfv~xCfx fCGCG
E~fECxCbvD|bfb6|GE/2]CfGxfDxvxffECvGb~xMfE3
bD/2|vfCCxvGGD]xC DMCMMMbfCxbMDvfGxfGC3:|
ECEeEMfvMCEvC bvCGEx ~GffEM4|:GxvGMb~E
vCM~~CG~xbbbCxvbCxG MxfMvvD/2DxxEEGvvG
vf~GbxbxE XXf X/2:|~bbCExb~ME~bGGMGCf xbG
CfMCCC xffbGxvGCbEbEfG~xv [ M[ [ EfbvC[ M[ C
[ BfbMMvGf~MbfGb~~xfb/2~bCbfbvffbeMbCGCCx
Cb]/2EffEbbx/2GfvCEGf/2::MEv] ]fxb/2vEG~xxxM
fExCMb EECxGvff~Ebb4bE~vGfbxMxxfM~fX/2|
:ffffMMMGxDxDvMMEC Mvxx3:|v~ fbECbEfCxGx6|
```

Invalid Key Prediction

Fig. 17. Model A at 1000 epochs

```
X:1
T:PREDICTION
M:6/8
K:G
Q:1/2=60
V:1
%%MIDI program 0
EGEEBd|"G"BAFAdc1|"D"dzadzf|zzb4|
b'AAAed::dbbe/2cE+E/2::zEdEzd|bG+c4|dG+cded1|
"E"cbzz3|zbEzbe|ebeebb|"F"Ebzb/2Eb1z/2|zEbG+z2|
G+ebEzz|E+b5|AA+3z2|zbcze1|"G"G+GGG3|z3zbz1|
EEbbbE,|zGG,G+G3|dbf3z1|dAA'3d1|d3zEb1|zEz3c|
dAb3c|cacdcg|bcdbdc|ede/2zEzE/2|zdeG+eb|EzE+G+zE|
zEdzCz1|GG+G3d1|G,G+G3z|Ez3b2|e6|bE+bebe|
dbzE+b2|bdbgdb1|
```

Repetitive musical notes

Fig. 18. PRECON-LSTM at 1000 epochs

not learned the textual note patterns of notes properly as of now. The music generated by the model, when listened to, is found to be unpleasant.

2) *At 1000 epochs*

At 1000 epochs, it can be observed (fig. 17) that Model A has started to learn the structure and syntax of the ABC notation. It has learned that the key header is to be predicted after the meter header. However, the key is not predicted correctly here. The model is now making better predictions of musical notes in comparison to Model A at 500 epochs.

PRECON-LSTM at 1000 epochs (fig. 18), just like Model A, is able to produce syntactically correct music because of the employed reconstruction techniques. The model is even predicting musical notes in a better manner, with much more variety. Still, a small amount of musical notes repetition exists. Upon hearing the generated music, it is found to be pleasant initially, but loses quality after some time.

3) *At 1500 epochs*

Model A has learned the syntax of the ABC notation. It is able to predict the headers, as well as the comments occurring in the ABC notation. In fig. 19, it can be seen that the model has predicted the '*K*' header correctly, as well as the comment '%Nottingham Music Database'. This model is also able to generate music without any reconstruction techniques. It is however, yet to predict the musical notes in a better manner. The music generated using the generated ABC notation is found to be unpleasant when listened to.

```
X:1
T:PREDICTION
M:6/8
K:G
% Nottingham Music Database
EEd2dcd|"A"cB2BBB|"B"zBBdBD|F_D^d^e|De/2z/2d^d|
d4|z/2C/2_bb'z|DD+DF+e/2e/4Dfd/4|BB5|zz2zde|ddFCEE,|
b'z^B4|aa/2g/2f/2e/2d/2EA/2|cBcA|G4|Bc3|c'ee/2df/2|
Bfd^d|^Cd_bb|^f/2fGzf/2|f^d2d:|z/4z^DFD3/4|FDFz1|
D+C^GB|DDeD1|"G"^d^ff/2f/2_f|^d^d_d^d|^CdC2|
e/2d^dCE/2|d_bbz|"F"^B_dB_B|^f^fC^d|dCz^B|
"A"Cd_B,d|"E"zz3|:f^B^f2|_B2z2|GBf2|f^fC^d|^F_fz^d|
^dC^f^f|Cb'b'd1|zdde/4de_d3/4|GB+DeGD|eGD^GB+C|
^G4d^f|^dCE+CzD|e/2d/2eDe3|e/2e_e^d_D2e/2|
_D^d^e_Ded1|e/2e/2_D5|eD,^d_Dz^e/2d/2|d4|z4|z4|
zzfd1:|z^B2z1|GB/2ee^dCd/2|z^dd/2De/2d^e|f4|
```

Not consistent with no. of beats per measure specified in meter

Fig. 19. Model A at 1500 epochs

```
X:1
T:PREDICTION
M:6/8
K:G
Q:1/2=60
V:1
%%MIDI program 0
EGEEBd|"G"BAFAde|"D"cdeceb|"B"ebbdbc|E,E,e4|
de3cd|cze4|EdE+b3|b6|gbcc3|bdbzd/2b/2a|b6|
EdG+3G|g6|EbEE+3|g3b/2bEb/2|dbzbgE|dbdbG+d|
de3bd|zEdzE+b|bG+zEbe1|G+cebEb|E,beG+ce|"C"ce5|
eceE,bc|"G"egbE+be|gcdceE+1|zEbcee|"G"beG+bzz|
E+bbEzb|G+6|Ebedbd|ee5:|dEdbdG|+bzzz3|G+G5|
dEdbbd::E+3zbz1|:dddddc|"B"dGEb3:|"A"bzzdb2|
z3cce|b6|
```

Fig. 20. PRECON-LSTM at 1500 epochs

Fig. 20 shows that PRECON-LSTM at 1500 epochs is able to produce a variety of musical notes. The music generated, when heard, is found to be very pleasant, and does not lose quality even after some time. Just like the previous models, this model too generates syntactically correct ABC notation.

V. CONCLUSION

In this paper, we present an LSTM based model, called PRECON-LSTM, with some effective dataset preprocessing and reconstruction techniques for the purpose of music generation. The model uses ABC notation, a textual form of music, for representing music here. The model tries to learn the pattern of musical notes, and uses character level modeling to predict the musical notes. Some of the selected pieces of musical tunes artificially generated by PRECON-LSTM are available at *https://soundcloud.com/sarthak-agarwal/sets/ai-music*.

On subjective evaluation by peers, it was found that the proposed model is able to generate pleasant and good quality music. The model is able to generate music which is very similar to music composed by humans. Experiments suggest that the model is able to learn the different patterns occurring in the textual representation of musical notes.

The dataset preprocessing and reconstruction techniques employed have proven to be advantageous when compared to a model not employing such techniques. These techniques allow the model to focus on learning the musical notes only, with the structure and syntax of the ABC notation being reconstructed later. Using such techniques reduce the training time, and provide better prediction of musical notes for the same number of epochs, in comparison to a model without such techniques. Also, proper reconstruction techniques ensure the generation of syntactically correct music in ABC notation.

The current work focuses on the generation of monophonic music only. In future, this work could be extended to produce polyphonic music as well.

REFERENCES

[1] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". Neural Computation, 9(8):17351780, 1997.

[2] How to understand ABC (the basics), http://abcnotation.com/blog/2010/01/31/how-to-understand-abc-the-basics/

[3] I. Xenakis. "Formalized Music. Thoughts and Mathematics in Composition.". Indiana University Press, 1971.

[4] A. Horner and D. E. Goldberg. "Genetic Algorithms and Computer Assisted Music Composition.". In Proceedings of the Fourth International Conference on Genetic Algorithms, 1971.

[5] P. M. Todd. "A Connectionist Approach to Algorithmic Composition". Computer Music Journal, 13 (4):2743, 1989.

[6] M. C. Mozer. "Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing". Connection Science, 6 (2-3): 247280, 1994.

[7] D. Eck and J. Schmidhuber. "A First Look at Music Composition using LSTM Recurrent Neural Networks". Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Technical Report No. IDSIA-07-02, 2002.

[8] The Unreasonable Effectiveness of Recurrent Neural Networks, https://karpathy.github.io/2015/05/21/rnn-effectiveness/

[9] L. Johnston. "Using LSTM Recurrent Neural Networks for Music Generation". 2016.

[10] K. Huang, Q. Jung and J. Lu. "Algorithmic Music Composition using Recurrent Neural Networking". 2017.

[11] B. Sturm, J. Santos and I. Korshunova. "Folk Music Style modeling by Recurrent Neural Networks with Long Short Term Memory Units". 16th International Society for Music Information Retrieval Conference, late-breaking demo session, 2015.

[12] ABC version of the Nottingham Music Database, http://abc.sourceforge.net/NMD/

[13] abcmidi, https://github.com/leesavide/abcmidi

[14] J. Duchi, E. Hazan and Y. Singer. "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". Journal of Machine Learning Research, vol. 12, pp. 21212159, 2011.

[15] P. de-Boer, D. Kroese, S. Mannor and R. Rubinstein. "A tutorial on the cross-entropy method". Annals of Operations Research, 134, 1967, 2005.

[16] M. C. Mozer. "Neural network composition by prediction: Exploring the benefits of psychophysical constraints and multiscale processing". Cognitive Science, 6:247-280, 1994.