Machine Learning Translation from Arab Vocal Improvisation to Instrumental Melodic Accompaniment

By
Aya Mahmood Alshamaileh

Supervisor
Dr. Gheith Ali Abandah, Prof

This Thesis was submitted in Partial Fulfillment of the Requirements for the Master's Degree of Science in Computer Engineering and Networks

School of Graduate Studies
The University of Jordan

Dec, 2019

COMMITTEE DECISION

This Thesis (Machine Learning Translation From Arab Vocal Improvisation To Instrumental Melodic Accompaniment) was successfully defended and approved on----------

Examination Committee	**Signature**
~~Prof~~Dr. Gheith Abandah~~,~~ (Supervisor) Prof. of Computer Architecture	----------
Dr. Ali Alhaj (Member) ~~Assoc.~~ Prof. of Computer Architecture	----------
Dr. Iyad Jafar~~,~~ (Member) ~~Assoc.~~ Prof. of Digital Image Processing	----------
Dr. Mohammad Abd-Almajeed (Member) ~~Assoc~~Assistant~~.~~ Prof. of Computer Architecture	----------

DEDICATION

To my parents for their prayers and endless support

To my beloved husband

To my little boys Hussein and Hashem

To my sister Eman

ACKNOWLEDGEMENT

I would like to thank my supervisor Prof. Gheith Abandah for his advices, suggestions, help and patience during our work on this thesis. This project would not have been possible without you, I am grateful to you. You made a great effort with me to achieve this work in a correct way.

I am grateful to ~~Dr~~Prof. Kamel Ismaili ~~-~~and Eng. Fadi AlGawanmeh for their cooperation and ~~assistance in collecting the data~~sharing and explaining ~~it to~~their dataset with me.

Formatted: Line spacing: Double

~~List~~ Table of Contents

Commented [GA1]: Use page breaks, instead of several new lines, to make a section start on a new page.

List of Figures

List of Tables

List of Abbreviations

Commented [GA2]: Sort this list alphabetically.

MLT	Machine Learning Translation
BLEU	Bilingual Evaluation Understudy
MIR	Music Information Retrieval
LSTM	Long-Short Time Memory
MIDI	Musical Instrument Digital Interface
RNN	Recurrent Neural Network
ANN_S	Artificial Neural Networks
MAE	Mean Absolute Error
MSE	Mean Square Error
GRU	Gated Recurrent Layer
DNN	Deep Neural Network
RMSPROP	Root Mean Square Propagation
AMT	Automatic Music Transcription
SMT	Statistical Machine Translation
API	Applications Programming Interfaces
CSV	Comma-Separated Values
MT	Machine Translation

Machine Learning Translation from Arab Vocal Improvisation to Instrumental Melodic Accompaniment

Formatted: Left-to-right, Indent: First line: 0"

By
Aya Mahmood Alshamaileh

Supervisor
Dr. Gheith Ali Abandah, Prof.

Formatted: Centered, Indent: First line: 0"

ABSTRACT

Formatted: Indent: First line: 0.49", Space After: 0 pt, Line spacing: Double

Vocal improvisation is a traditional musical form of Arab music called *Mwwal*. It is a non-metric form accompanied by short instrumental sentence composition. This type of music is popular in the Middle East and Africa. Despite that, few researchers worked on accompaniment systems of Arabic music. The ~~western~~ Western harmonic accompaniment patterns for various musical forms such as Jazz and chorale styles have grown rapidly compared with Arabic music.

Making music accompaniment system is an important and worth step in Arabic music, many researchers have developed applications and tools to automatically accompany Arabic vocal improvisation to their instrumental melodic tones to expand the use of creating Arabic music by non-musicians people easily without any experience in generating music.

Many systems have been developed by the researchers to solve the automatic accompaniment problems using real-time music systems, rule-based systems, and

statistical machine translation systems which guided us to look forward to using Machine Learning Translation (MLT) system to solve the problem.

Commented [GA3]: Add here some details about used dataset and network: dataset size, RNN, LSTM, E/D, and attention.

The statistical machine translation approach that is used to solve the accompaniment problem was measured by the Bilingual Evaluation Understudy (BLEU) metric algorithm, which tested the quality of the output results with human judgment in real life. We used the same metric algorithm to test the efficiency of the model that we proposed on the same data corpus using machine learning translation.

Our proposed system achieves state-of-the-art results over the best reported results in the same field. They got 24.62 from ~~Vocal~~ vocal to instrumental and 24.07 from instrumental to vocal in the BLEU test. Our system increases the BLEU result to 60.~~5~~6% ~~from~~ for vocal to instrumentals and 40.5% ~~from~~ for instrumentals to vocal, which is good improvement compared with ~~over~~ the ~~best~~ previous published result.

CHAPTER I

Introduction

This chapter ~~will~~ discusses the definition of Arab vocal improvisation, note representation and note features, MIDI file, importance of Arab vocal improvisation, research objectives, and the thesis outlines.

1.1 Arab Vocal Improvisation

Vocal improvisation is ~~the~~ a way that the singer performs a song differently to fit into his/her singing style and usually it is considered as an introduction to a song (Larson, ~~S.~~ 2005). Arab vocal improvisation is called *Mwwal*. It is a traditional type of music in the Middle East and North Africa. *Mawaweel* (plural of *Mwwal*) based on the singer's sense which called *Saltanah* (modal ecstasy) with short instrumental melodic sentences. The instrumentalist helps the singer to improvise the Mwwal according to the specified *Maqam* or *Taqsim* (Al-Ghawanmeh, et al., 2016). The singer of *Mwwal* moves between vocal sentences ~~by~~ and long rests while the instrumentalist played a musical sentence to fill up the gap that happened during the singer silence.

The main difference~~s~~ between Arab vocal improvisation and Western vocal improvisation is that Western improvisation is based on a scale or regular harmony; in other words, western style plays a certain key that restricts the singer behavior to be within the specified harmony. Arab vocal improvisation is a non-harmonized music style. It is based on a given *Maqam* which gives the singer more chances and independence on stage by interacting with the audience. Simply the difference can be sensed by ear, where the moves between *Maqamat* (plural of *Maqam*) are different in each style. The similarity between Western and Arab vocal improvisation styles is *Taqsim* which is the art of

improvisation.

1.2 Note Representation and Note Features~~: Degree and Duration~~

Musical sentences have many features such as loudness, timbre, melody, rhythm, pitch, harmony, and duration~~, .~~ ~~these~~ These notes are usually represented in a musical sheet with black dots and tails. The grid that holds these notes is called staff, which consists of horizontal lines where the notes are written vertically ~~regarding~~ according to their pitches (degrees). The vertical lines divide the staff into measures which ~~denoting~~ denote the duration signature as shown in Figure1.

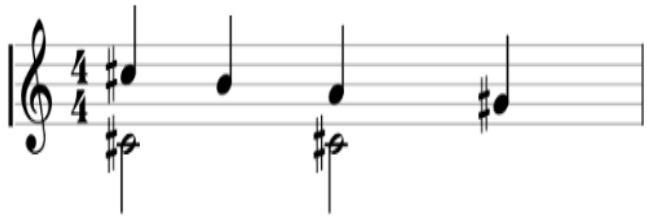


Figure 1: ~~Staff~~ Musical staff with notes

The main two features used in this work are pitch (degree) and duration; they are represented as scaled degrees and quantized durations respectively. ~~Degree~~ The degree denotes the location of the scale on the musical instrument and it is described using frequencies scaled to be between -7 and 7. Duration is the period or the ~~distance~~ time between each degree and the other and it is a positive number.

Choosing the correct musical note feature was a challenge because of the variety of them between each musical note to another. For instance, loudness is a non-metric feature and it is measured subjectively and it ranges between quite to loud that are inefficient to be measured as a numerical value. Timber feature refers to the speed and the quality of the musical note and that is also measured subjectively, on the other hand,

degree and duration features could be represented as numerical values and most of the researchers used them as a musical interface.

1.3 MIDI File

Musical Instrument Digital Interface (MIDI) is an interface to present the musical sheets digitally, the idea is to encode the properties of notes as numbers; the main musical properties which encoded are degrees and durations. The degree of the note recorded from the sending commands to play or stop a note which are represented as a scaled format. Also, the timing of these notes encoded by calculating the difference between the previous note and the current note time, MIDI files has a lot of formats and structures that allows us to easily extract and play musical information (de Oliveira, 2017).

1.4 Importance of Arab Vocal Improvisation

Arabic music has considerable aesthetics and it is worth to be publicized among the youth musician generations. Western musical forms present in several accompaniment models (Raphael, 2004). This is motivating us to enhance the computerized music generators to help musicians to create and correct their musical sheets referring to confidential reference. Also, we are looking for expanding the use of creating musical tones by non-musicians people easily without any experience in generating music (Al-Ghawanmeh, 2012). We will standrely on the fact that the iInstrumentalists interact with the singer by playing along with him or her throughout every vocal sentence by recapitulating that sentence instrumentally upon its completion (Racy, 1998). This fact will helps us in this work to find the best accompaniment system between any vocal and instrumental sequences.

Few researchers wrote about translation translating Arab vocal improvisation to instrumental melodic accompaniment. The western harmonic accompaniment patterns for

Commented [GA4]: Not clear. Refrase

various musical forms such as Jazz and chorale styles have grown rapidly compared to Arabic music (Hidaka, 1995).

1.5 Research Objectives

The growth of neural networks ~~is~~ has encouraged us to ~~be~~ use~~d~~ them in this domain to help musician and non-musician people when attempting to accompany vocal sentences to their corresponded musical sentence. Using Recurrent Neural Network (RNN) is our objective in this thesis to find a good approach to solve this problem better than the other methods.

Our proposed approach uses supervised RNN to map the vocal input sentences to instrumental output sentences. We used encoder-decoder RNN model that entered vocal sentences as input sequence, and the target sequence is MIDI sentences. Both of these two files contain pairs of degrees and durations with variable sequence length within one sentence.

The objectives and contributions of this project are summarized as follows:

- Investigating the use of machine learning algorithms using TensorFlow with Keras backend. Keras is a high level API built on ~~the~~ top of TensorFlow framework that is ~~exploded~~ maintained by Google's brain team ~~in~~ since 2015 (Gulli, *et al*., 2017). It is open source software that can be used in deep machine learning ~~algorithms~~tasks. Also, it is a python library that allows users to do experiments easily to debug and clearly to find user errors (Chollet, F. 2016).
- Building an accurate system by training it on a corpus of vocal improvisation sentences that were recorded manually by (Al-Ghawanmeh

and Smaıli, 2017). The corpus size is 2779 sentences of parallel vocal and instrumental sequences. We aim to train the system to achieve a better accompaniment system than the previous published systems in the same domain.

- Test the performance of our system using BLEU measure and compare it with the previous results. This measure is used to evaluate the accuracy in a bidirectional way (Papineni, 2002). Our goal is to exceed 24.62 from ~~Vocal~~ vocal to instrumental and 24.07 from instrumental to vocal in the BLEU test.

We hope that this thesis ~~will~~ contributes to develop the automated accompaniment systems of Arab vocal improvisation with the best instrumental melodic and give a good and helpful approach that could be represented in the future as an application with better accuracy subjectively and objectively than the other applications.

1.6 Thesis Outline

The rest of this thesis is structured as follows. Chapter 2 ~~re~~presents a literature review of previous related approaches that were developed to solve the accompaniment problem. Chapter 3 explains the technologies used in our work including TensorFlow and Keras, Recurrent Neural Network (RNN), Long-Short Term Memory (LSTM), encoder-decoder technology, Deep Neural Network (DNN) and attention. Also, Chapter 4 describes the methodology of our work which includes an explanation about the used data, data pre-processing steps, the use of masking layer, data encoding, inference model, and evaluation metrics. Chapter 5 describes the experiments that we ~~do~~ did in this work and how we ~~did tuning for~~tuned the proposed model and the results of our proposed system. Finally, ~~chapter~~ Capter 6 ~~shows~~ gives the conclusions and suggestions for future

work.

CHAPTER II

Literature Review

~~Musical~~ Auomatic musical accompaniment systems started from the mid-eighties, automating the musical accompaniment systems is a research area which leads to providing applications that can be used by non-musicians. MySong is an automated system published in 2008 that chooses musical melodies to accompany a vocal melody entered by a microphone the results are good subjectively for pop music and jazz (Simon, *et al.,* 2008). Another automated software application that is popular for doing accompaniment to generate powerful and creative music compositions is called Band In a Box. This application can be downloaded easily on Android and iPhone (Wilkins and Dennis, 2017). Also, there is a website that provided an automatic accompaniment to Arab vocal improvisation service called *Mawaweel* website (Al-Ghawanmeh, *et al*., 2015). Western music researchers enhanced the melodic accompaniment system faster than Eastern music ~~that is leading to look for their achievements in this field~~. This part ~~will~~ explains the best published three approaches used to improve the accompaniment problem. These approaches are: live solo composition (Dannenberg, 1984), Music Information Retrieval (Haddad, *et al*., 2010), and Statistical Machine Translation (Smaıli and Al-Ghawanmeh, 2017).

These approaches ~~will~~ are discussed in the following sections.

2.1 Live Solo Composition

Live solo composition constructed by a solo singer as an input of the model and the output is the match entered musical sentences with the best notes given by computers. This technique can recover the singer mistakes and reconnect the accompaniment between the solo singer and the detected musical degrees.

Commented [GA5]: Not clear, rephrase.

The system achieved good results even if the degree of the note varieties suddenly by adding or deleting additional sentences by the singer. However, the two main bottlenecks of this system are that the system did not use any timing information input entered into the model and that leads to low accuracy and overlapping the notes occurs by neglecting the duration between notes. Also, they assumed that the music player did order set of events on his musical instrument, so the matching considered a temporal order of events.

Commented [GA6]: Not clear, rephrase.

Dannenberg used an on-line algorithm to build matrices for input and determine the best length that should be taken to evaluate the output. Sequence length increments regarding the entered input are given by live singer. When the singer stops; the matrix of the best length used to evaluate the output scores accompanies the inputs (Dannenberg, 1984).

2.2 Music Information Retrieval (MIR)

This approach considered as a new theory in Arabian music aspects, the importance of the approaches that focus on Arabic music refers to the lack of previous studies of this music style.

Commented [GA7]: Not clear

Music Information Retrieval (MIR) is a sequential module connected to sequential phases. These phases do~~ing~~ segmentation, frequency detection, filtration, pitch decision, and MIDI Matrix creation. ~~in (2010)~~ MIR system was used as educational tool~~s~~ for woodwinds like Nay and *Shababeh* the system was proposed by researchers from the

University of Jordan to provide an opportunity to apply music in e-learning. The constructed MIR system ~~evaluated~~ includes three different applications; MIR for Automatic Music Transcription (AMT) of Arabian Woodwinds, MIR to serve Arabian composition, and "query-by-playing" for Musical Libraries (Haddad, *et al*., 2010).

Automatic Music Transcription: ~~the~~ The system receives a vocal file as input and generates a full score output. This system deals with Nay and *Shababeh* instruments only and this is one of the system ~~bottlenecks~~limitations. MIR to serve Arabian composition is an application used to help Arabian polyphonic. ~~'~~*Query by-playing*~~'~~ application is considered a great development ~~in~~ of the music department at the University of Jordan. Melody retrieval systems are based on querying the input as a vocal file created by a user who sings or plays a short song into a microphone, then the system looks for the best match~~ed~~ of musical notes based on a reference musical transcription~~s~~. MIR Developed music through applications that help musicians in correcting musical errors and gives hints to musicians while training. Nowadays dynamic MIR is a search domain.

2.3 Statistical Machine Translation (SMT)

This approach ~~used to make~~developed an accompaniment system between Arab vocal improvisation~~s~~ which is defined as singing narrative poetry on a given *Maqam* called *Mwwal*~~,~~. ~~the~~ The resultant song should be recapitulated by an instrumentalist who accompanies the vocal sentence with the corresponding melodic notes upon its completion (Smaıli and Al-Ghawanmeh, 2017). The researchers of this work used transcriptions recorded by their instrumentalists and singers in closed rooms to construct their database so they ensured that the data have no noise due to the absence of the audience.

The used corpus consists of 2779 parallel vocal and instrumental sentences. Statistics applied on this corpus which was recorded manually sentence by sentence are shown in Table1.

Table 1: Statistics on parallel corpus

Measure	Vocal	Instrumental
Total number of sentences ~~on whole corpus~~	2779	2779
Total duration on whole corpus	17907s	13787s
Note count on whole corpus	35745	55665
Maximum note count within one sentence	82	142
Minimum note count within one sentence	1	1

The vocal sentences are longer than instrumental sentences, although the instrumental notes are ~~bigger~~more frequent; ~~that's referred~~this is due to the fact that the instrumental sentences imitated with the picked off keys generated by the instrument. The sound of the key does not tolerate for a long time unless the instrumental keeps plucking the keyboard sound for long. The table also shows that for both vocal and instrumental sentences the length of the note (degree) can be as a maximum ~~to~~ 82 and 142 vocal and instrumental sentences, respectively, and one note for both sentences as a minimum.

The musical note consists of varying features. This paper focused on the degree and duration features only. They represent them as a scaled degree by converting the note letters to a scaled number for simplicity and quantized duration by plotting both vocal and instrumental histograms to find the best quantized steps and duration ranges between

them. Fewer quantization steps give better translation results for both vocal and instrumental sequences.

The experimental part shows that the researchers applied statistics in Machine Translation (MT). They used the MT system to ~~built~~ build the accompaniment model. This model has different settings based on three variables. The first variable is the score reduction, which allows the user to reduce the music score to be shorter by reducing sustain informed by removing the unessential degrees and durations or reducing the silence by replacing each unessential note by a silence note that merged the durations of the unessential notes. Un-reducing the notes means that the score reduction variable does not change. The second variable is merging adjacent similar notes which including merging features of the two similar notes with one longer note to minimize the instrumental sentence or ~~Un~~un-merged feature means that this variable does not change. The final variable is how to represent the note including three representations classified as scaled degrees representation only or quantized durations only or both of scaled degrees and quantized durations.

The best results have been achieved by applying to merge the two similar notes, but without score reduction with a scaled degree and quantized duration representation. Testing the results of the BLEU measurement gives 24.62 from vocal to instrumental and 24.07 from instrumental to vocal.

In this thesis, we applied Machine Learning Translation (MLT) using RNN to ~~do~~ build accompaniment system for Arab vocal improvisation to instrumental melodic on the same corpus that was used in the statistical machine translation approach. We test the results with the BLEU score which improved over state-of-art statistical machine translation approach.

CHAPTER III

Technologies Used

This chapter ~~demonstrates~~ describes the most important technologies used in this work. The following sections are TensorFlow and Keras, Recurrent Neural Networks (RNN), Long-Short Term Memory, encoder-decoder technology, Deep Neural Network (DNN), and attention.

Formatted: Font: (Default) +Headings CS (Times New Roman), Complex Script Font: +Headings CS (Times New Roman)

Formatted: Font: (Default) +Headings CS (Times New Roman), Complex Script Font: +Headings CS (Times New Roman)

3.1 TensorFlow and Keras

TensorFlow is an open-source library ~~that~~ provided by ~~a~~ Google brain team to be used in neural networks projects (Géron, 2017). It supports working with computational graphs, operations, tensors, and sessions. Also, it provides the programmer with ~~much~~ high-level Application Programming Interfaces (API) which are considered friendlier to use (Abadi, *et al*., 2016).

Keras is one of the embedded ~~APIS~~ APIs ~~to~~ of TensorFlow which is designed to help the programmer in building models, easy extending them, debugging, and training with less time and effort. Python is the language used in Keras. It is powerful and easy to understand (Gulli and Pal, 2017).

Keras is classified in two main models sequential and functional APIs~~,~~. ~~the~~ The sequential model is a built-in model in Keras which we can import ~~it~~ from Keras libraries; it is produced in a linear stack pipeline layer with predefined models like dense layers, ~~or~~ Gated Recurrent Unit (GRU) layers, and ~~or~~ Long-Short Term Memory (LSTM) layers. On the other hand, the functional model can be used in complex tasks or shared layers with multi outputs, the layer could be GRU or LSTM as in the sequential model, and the model used for training and validation phases are generated by the user, unlike the previously mentioned model. Both of the two models supporting predefined activation, loss, optimizer and metric~~s~~ functions that help in fitting and compiling the model (Ketkar, ~~N.~~ 2017).

This work uses a Keras functional model to deal with multi outputs using Mean Absolute Error (MAE) loss function, Root Mean Square Prorogation (RMSPROP) optimizer function and the accuracy metric.

3.2 Recurrent Neural Networks (RNN)

Recurrent neural network~~s~~ is a ~~part~~ type of Artificial Neural Networks (ANNs) that ~~performed~~ performs in a similar way of human brain performance using many neurons to save information related to a sequence of data that is entered to the system, the saved data in RNN are considered as memories in the human brain which can be collected any time from the neurons and retrieve or guessing what will ~~be~~ happened if there is a sudden change in these data in the same situation. This innovation helps in solving problems that need a human brain for learning and making decisions, Figure 2 ~~presenting~~ presents the structure of RNN (Svozil, *et al*., 1997).

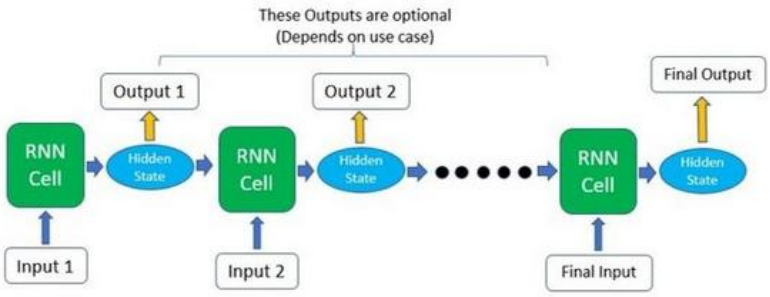


Figure 2: RNN structure

Formatted: Centered, Indent: First line: 0"

~~The classification of RNN types depends on the factor of this classification that means if we use the presence of data as a factor, RNN classified with three types: supervised RNN, unsupervised RNN, and semi supervised RNN. The supervised type is where there is a sequence input x and a sequence output y, and the RNN used to train the neural network to find the mapping between them, so if there is new input data the mapped function can predict the output y. in unsupervised type only the input data x is available and there is no information about output data y. the semi supervised type falls in between the two mentioned types were some of the data known and others are not.~~

Commented [GA8]: This applies to machine learning in general, no just RNNs

The ~~second classification of~~ RNNs ~~determines~~ are classified based on the input and output sequence lengths. Figure 3 ~~explained the variation of length~~shows this classification (Svozil, *et al*., 1997).

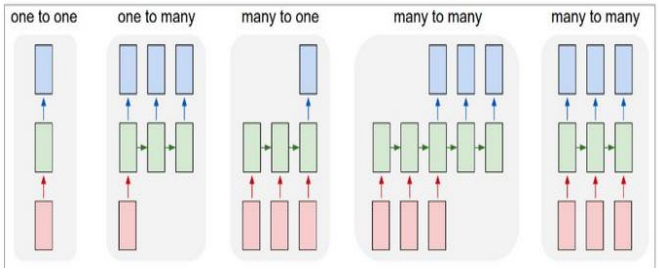


Figure 3: Input and output sequence lengths of RNN

The red rectangles are the input data, the blue is the output and the green ~~is~~ are the states. Each rectangle represents a vector and the arrows are functions applied to these vectors. From left to right: one-to-~~-~~one is the vanilla ~~mode~~type, ~~which~~ where RNN is not applied, one-to-many receives single input like image, and the output is a sequence like explanation ~~of each~~or caption of ~~this~~ the image~~.~~, In many-to-~~-~~one, the input is an explanation of a task and the output is a ~~Boolean true or false~~vector not a sequence, the fourth diagram is many-to-~~-~~many, which ~~included~~ includes variable sequences of inputs and outputs like machine translation from one language to another. The last one is many-to-~~-~~many with a fixed length, this type used in labeling the entered videos for each frame.

This thesis ~~works on~~uses RNN with supervised ~~type mode~~learning and many-to-many ~~of~~ variable sequence length, in which the input is vocal sentences and the output is MIDI sentences with variable lengths for each sentence ~~in both~~. RNN should map the input and the output and finding the best learning algorithm depending on the training data derived from the whole corpus, then we reconstruct the algorithm to predict the target sequence regarding the validation data and the results justified by the machine learning performance metrics. The training is stopped where the validation loss does not improve to avoid the ~~Overfitting~~ overfitting problem. ~~The~~ Overfitting occurs when the training loss decreases and the validation loss increases due to the noisy data or the leakage of the corpus samples; Figure 4 shows the ~~Overfitting~~ overfitting problem (Gulli, *et al*., 2017).

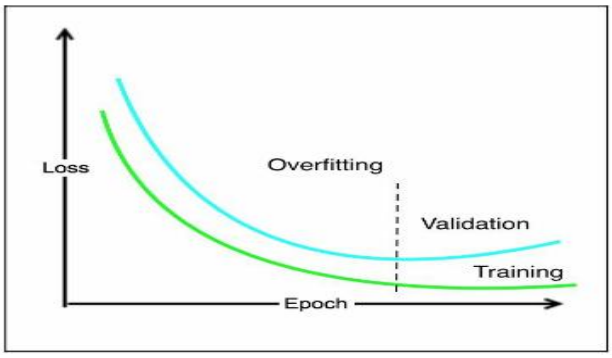


Figure 4: Overfitting problem

3.3 Long-Short Term Memory (LSTM)

The long short-term memory network is a special recurrent neural network, which can learn long term dependencies; ~~on other meaning~~ it can remember information for a long time using memory cells.

The LSTM cell was created to store information from previous context, ~~LSTM cell~~it consists of three gates: input gate, forget gate and ~~the~~ output gate as shown in ~~figure~~ Figure 5 (Graves and Schmidhuber, 2005).

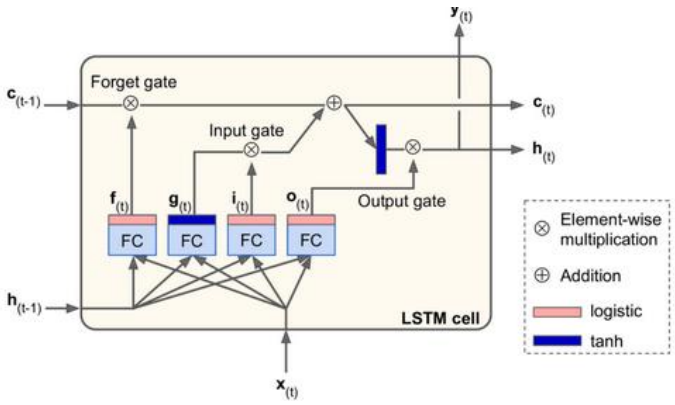


Figure 5: LSTM cell

~~The three main gates in LSTM cell is, the~~ The input gate~~, which it~~ receives the input data and the short term memory hidden state of the previous cell, then it ~~is~~ analyzes the current input and the state to control the output of the main layer $g_{(t)}$~~,~~. ~~the~~ The forget gate~~,~~ learns the network what to store in the long term state $c_{(t-1)}$, or what to throw away, and what to read from it, so at each time step some memories are dropped and some memories are added with the input gate. The output gate filter~~ed~~s the results of the previous gates and controls which parts of long term memories should be read from the short term hidden state (Hochreiter and Schmidhuber, 1997).

3.4 Encoder Decoder ~~Technology~~Architecture

Commented [GA9]: Don't forget to update the Table on Contents

The encoder-decoder model is a recurrent neural network that deals with variable sequence lengths. It consists of two parts; the encoder ~~it~~ that is a sequence to vector network, which takes the input data and encode~~d~~s it to produce outputs and states~~,~~. ~~these~~ These states pass~~ed~~ to the second part which is a vector to sequence network called decoder, the decoder takes the states of the encoder and use~~d~~s them in training the network to generate the target sequence as shown in Figure 6 (Cho, K., *et al*., 2014).

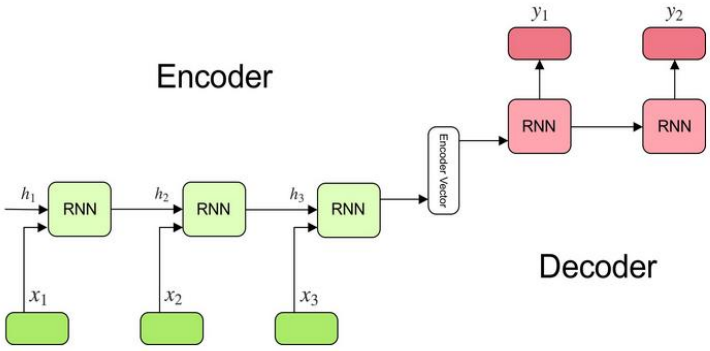


Figure 6: Encoder decoder architecture

3.5 Deep Neural Network (DNN)

The hierarchal recurrent neural network of more than two layers is called Deep Neural Network (DNN). An ~~means that the~~ RNN ~~consists~~ consisting of multiple hidden layers~~,~~ each of them handles a part of the task problem and the output states of each layer ~~passed~~ passes to the next time step of the current layer and the next layer of the current time step as shown in Figure 7 (Kaiming, H, *et al*., 2016).

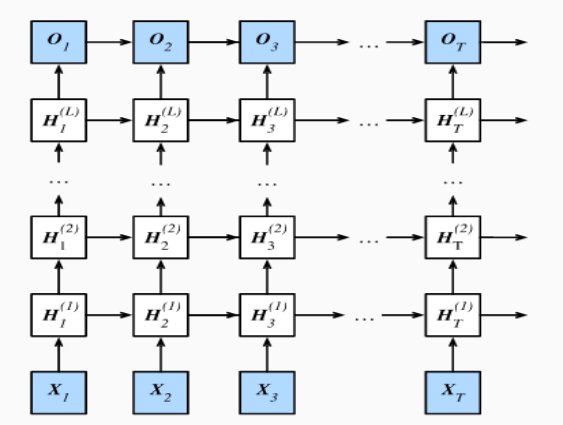


Figure 7: Architecture of deep neural network

Formatted: Centered, Indent: First line: 0"

The Deep Neural Network has proven ~~their~~ its ability to deal with sequence prediction problems such as machine translation (Zhang and Zong 2015)~~,~~ and speech recognition (Graves, A. *et al*., 2013~~),).~~ ~~it~~ It is ~~reliable to handle~~efficient in handling complicated problems with non-linear relationships, each layer in the DNN is a recurrent neural network, which handles a part of the problem to solve it, the higher layers deal with difficult and hard data and functions regarding to features extended from the input features of lower layers.

The input vector is X_i ~~as~~ shown in the ~~above~~ figure above and the hidden state~~s~~ vector~~s~~ ~~is~~ H_i which connect~~ing~~ the input layer with the output layer O_i according to specific hidden layer function.

3.6 Attention

The encoder ~~is the left part of the model which~~ encodes the whole row sequence and generates fixed-length vectors~~, on the validation stage;~~. ~~the~~ The decoder uses these fixed-length vectors to decode new input sequences. Researchers found that the fixed encoded vector is a performance bottleneck caused by using hidden layers with a lower number of neurons that compressed the encoded features~~,~~. ~~so we~~ We need an attention layer to solve the problem. The mechanism of attention stands on allowing the decoder to pay attention to some features in the input sequence row or sequences length of the input and output when predicting new output~~,~~. ~~so the~~ The decoder ~~focused~~ focuses on the desired input only without paying attention to the complete encoded sequence row. The new model creates new fixed-length vectors with the most important information for each time step prediction. This model ~~keeps the~~good performance ~~good~~, even if the translated sequences become longer than the actual sequences (Bahdanau, *et al*., 2014).

The two main types of attention layers are: global (soft attention) and local attention (hard attention~~),~~). ~~both~~ Both ~~of them produces as an~~are interface layers between the input and output layers.

- **Global attention** also called Bahdanau's attention. The procedure of this attention refers to additive attention. This attention is placed on all the input sequence positions. So, all encoder hidden states are considered for

Formatted: Font: Bold, Complex Script Font: Bold

generating the attention context vector. The algorithm steps are ~~described~~ as follows.

1. The procedure stands on the assumption that at time *t* we should be attended at the decoder hidden states at time *t*-1.

2. Calculate the attention scores for each encoder state using the previous decoder hidden states as follows:

$$attention\ score_t = W_{combined}(W_{decoder}^{t-1} \cdot H_{decoder}^{t-1} + W_{encoder}^{t} \cdot H_{encoder}^{t})$$

$$where, W^{t-1} = the\ previous\ weighted\ matrix.$$

$$H^{t-1} = the\ previous\ hidden\ state.$$

3. Calculating the context vector by multiplying the encoder output state with the attention scores as follows:

$$C_t = attention\ score_t \times H_{encoder}^{t}$$

4. The context vector is concatenated with the previous decoder hidden states and fed into the decoder RNN for that time step to produce new hidden state, which ~~used~~ uses to predict y_{t+1} as follows:

$$h_t = concat(C_t, h_{decoder}^{t-1})$$

5. Processes ~~from (~~2-4~~)~~ repeat until the vector exceeds the maximum sequence length.

- **Local attention** also called Luong's attention or multiplicative attention. The process of this mechanism differs from global attention in calculating the attention scores and the place of the attention layer in the input or output

Formatted: Font: Italic, Complex Script Font: Italic

Formatted: Font: Italic, Complex Script Font: Italic

Formatted: Font: Bold, Complex Script Font: Bold

layers~~,~~. ~~where the~~ The local attention is placed only on a few input sequence positions. In other words, only a few hidden states of the input are considered for generating the attention context vector. The algorithm procedure is similar to global attention, except attention score calculations, where in this attention the scores are calculated in two different ways as follows (Luong, *et al*., 2015).

1. Dot scoring function is a way to compute~~s~~ the attention weights by multiplying the hidden states of the input by the hidden states of the output:

$$attention\ score_t = (H^t_{decoder} \cdot H^t_{encoder})$$

2. Concatenation scoring function, it is similar to the way that the attention scores are calculated in global attention, except that in local attention they share~~d~~ the same weight~~ed~~ matrix and they use~~d~~ the current decoder hidden state instead of previous.

$$attention\ score_t = W_{combined}(H_{decoder} + H_{encoder})$$

Now after calculating the attention weights using one of the previous scoring functions, we can find the context vector similar to the global attention~~,~~. ~~then~~ Then the new hidden state is generated by concatenated the context vector with the current decoder hidden states, which are used to predict y_t as follows:

$$h_t = concat(C_t, h^t_{decoder})$$

The process repeats until the vector exceeds the maximum sequence length.

CHAPTER IV

Methodology

The accompaniment system needs vocal and musical transcriptions to train and test the model. So, we used the dataset that ~~(~~Al-Ghawanmeh ~~And~~ and Smaıli~~,~~ (2017) used in their statistical machine translation approach~~,~~. ~~the~~ The input data ~~are~~ is the vocal file and the target data ~~are~~ is the music file which is called MIDI file~~,~~. ~~these~~ These two files are used to train and test our model.

Our proposed system has two phases to accompany the vocal improvisation to their suitable musical notes: training and validation phases. In the training phase, we built LSTM of encoder-decoder RNN, the input data are training sets of vocal data which are encoded to get the target MIDI sentences by the decoder~~,~~. The preprocessing methods applied on the data ensure to remove or correct the unused or invalid sentences for both vocal and MIDI comma-separated value files (CSV). In the validation phase, we ~~re~~built the RNN to encode the input validation sequences to predict the target validation sequences.

Testing the reliability of the proposed system ~~should be using~~is done using the same metric used in statistical machine translation approach. So, we used Mean Square Error to test the loss of the data and BLEU score to compare the predicted data with the actual data ~~which considered as a good metric in translation systems on both sides~~.

4.1 Data

The experimental data were the same data used in (Smaıli and Al-Ghawanmeh 2017~~),~~). ~~the~~ The corpus consists of 2779 parallel sentences of vocal and MIDI data with variable sequence lengths~~,~~. ~~the~~ The minimum length could be one pair of both degree and duration, and the maximum length for both of vocal and MIDI are 82 and 140, respectively.

The total count number of sequences that their lengths are less than two is 11 for vocal data and 6 for MIDI data as shown in Table 2.

Table 2: Number of sequences of lengths less than two

Vocal	MIDI
11	6

These sequences do not add any information for the model in both the training and validation phases. Also, these ~~data~~ sequences are considered ~~as~~ redundant sequences because they are wasting the training time with low model utilization, so we ~~should~~ removed them to enhance the accuracy and training time of the model.

Each row sentence consists of multiple values of degree and duration as pairs~~,~~. ~~we~~ We ~~calculate~~ found the average duration of the entire sequence length for both of vocal and midi sequences as shown in ~~table~~ Table 3.

Table 3: Average duration per sequence length

Vocal	MIDI
6.4	5.0

Then we plot the histogram of the sequences lengths for both MIDI and vocal data to know the relationship between vocal and MIDI sentences as shown in Figure 8.

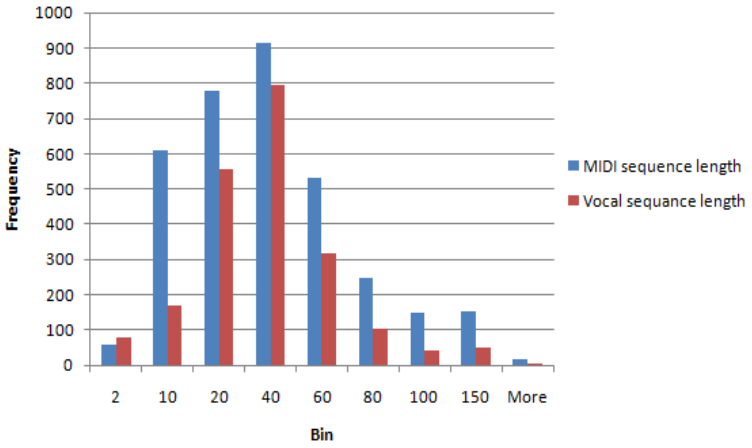


Figure 8: The histogram of MIDI and vocal sequence~~s~~ length~~s~~

We observe~~d~~ that the sequence length of MIDI data is larger than the vocal data, despite the average duration of MIDI is less than the vocal; due to the midi sequences belong to Oud style performance. Unlike in vocal music where the singer can sustain a musical note for as long as his breath can take, the Oud cannot. So, the instrumentalist needs to keep plucking the string to keep a particular sound running (each pluck is a new note). The result is having, on average, more notes in Oud than in vocal in a particular duration.

Plotting the histograms of durations in vocal and MIDI data ~~appear~~ reveals that most of durations surrounded between zero and one as shown in Figures 9 and ~~Figure~~ 10.

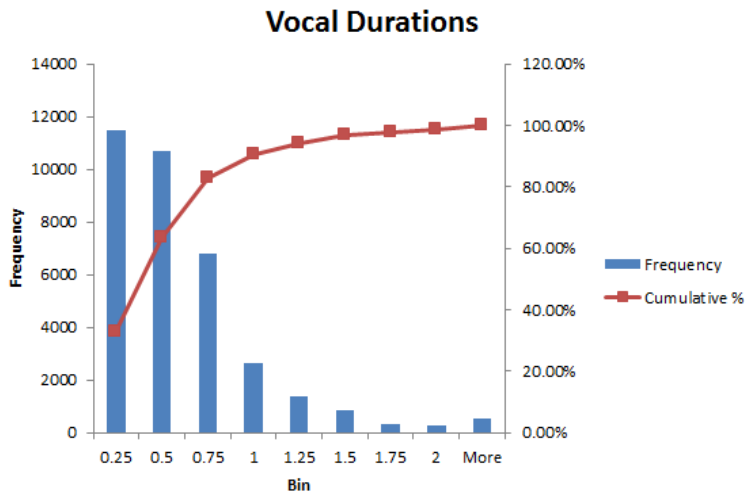


Figure 9: The histogram of vocal notes durations

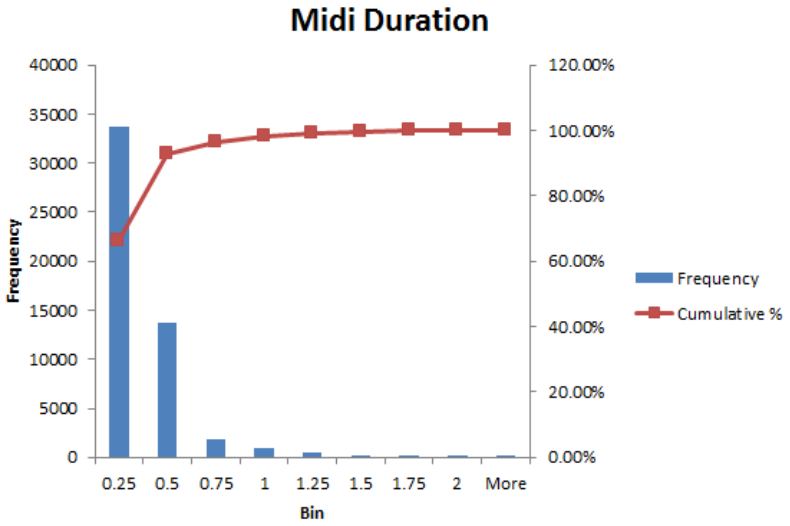


Figure 10: The histogram of MIDI notes durations

The degree of each of vocal or MIDI data should be within the range -7 to 7 and the durations are positive numbers when we check the degree values we found numbers that should not be appearing as shown in Table 4 and Table 5.

Table 4: Repetition of vocal note degrees

Vocal degree	Repetition
-7	144
-6	868
-5	963
-4	1073
-3	1209
-2	1702
-1	2441
0	196547
1	4754
2	5653
3	6129
4	3795
5	1902
6	382
7	82
93	**182**
100	**51**

Formatted: Font: Bold, Complex Script Font: Bold

Formatted: Font: Bold, Complex Script Font: Bold

Table 5: Repetition of MIDI note degrees

MIDI degree	Repetition
-7	2921
-6	872
-5	975
-4	1366
-3	2166
-2	1822
-1	2820
0	348335
1	8069
2	8688

3	7291
4	5116
5	2144
6	1082
7	513
93	**321**
100	**117**

Formatted: Font: Bold, Complex Script Font: Bold

Formatted: Font: Bold, Complex Script Font: Bold

These unexpected data (93 and 100) are caused by errors in ~~generated~~ generating the MIDI data in computers.~~,~~ ~~and it should be~~We replaced them in the pre-~~-~~processing stage to prevent bad performance.

4.2 Data Pre-processing

The pre-processing steps are done to handle the invalid data. First, we remove~~d~~ all the sentences shorter than two pairs. Second, the unexpected values like 93 and 100 are replaced with the median of the possible degrees, which is zero.

These modifications reduce the corpus from 2779 to 2762 sentences. After that, we randomly split the corpus to get 1933 parallel sentences for training and 829 for validation.

4.3 Masking ~~layer~~Layer

Masking layer is added to handle variable length inputs in RNN with encoder and decoder topology~~,~~ ~~-~~. ~~the~~ The encoder ~~encoded~~ encodes the input sequence to a fixed encoded vector; on the other hand, the decoder is used to decode the whole vector to generate the target output and this is a problem with variable sequences. The masking layer used to generate an array with maximum sequence length filled with a dummy

variable called masked value, and then it is multiplied with the input data, where it resizes the array with the actual input data and neglects the padded sequence.

4.4 Data Encoding

We prepared our data for RNN training and validation ~~experimental~~ experiments ~~part~~ by dividing the whole data into rows and columns~~,~~. ~~each~~ Each row ~~considered as~~is a sentence of one note~~,~~ and the columns ~~handle~~ has the degree and duration features as pairs. The input is variable-length vocal sentences~~,~~ and the output is also variable-length MIDI sentences. This hierarchal is important for building the supervised learning of RNN.

We use encoder- decoder LSTM layers to deal with the variable length sequences~~,~~ . ~~the~~ The encoder takes the vocal data and ~~encoded~~ encodes it to produce encoded vector, which consists of the encoder output and the encoder states. Usually, the encoded vector generated by the encoder has a fixed size length which is padded with an encoder internal padding layer~~,~~. ~~to~~ To solve this problem, we used the masking layer, which is explained in the previous section. The decoder takes the last encoder state~~s~~ as initial state~~s~~ and the target data as an input for the decoder layer, and then the model is trained to find the best mapping between the encoder and the decoder inputs. The generated model constructed by the inputs of both encoder and decoder layers, and the final output of the decoder layers, which is the target MIDI data.

4.5 Inference Model

The i~~I~~nference model, ~~in machine learning also called~~ predicting phase, ~~this phase~~is ~~re~~constructed ~~the model~~ using the encoder and decoder layers which are defined in the training phase. ~~to~~ To create the encoder model to enter the validation set as a new

input with respect to the weights that were generated by the trainable encoder, then the decoder model is redefined to predict the validation data set using the decoder trainable weights.

4.6 Evaluation Metrics

Evaluating machine learning models is an important part of any project; the results of the evaluation give information about the model performance, where it could give good results ~~in~~ using one metric and ~~pad~~ bad ~~in~~ using another. We ~~can classify these~~used the metrics ~~into three main sets~~: accuracy, loss functions, and BLEU score.

4.6.1 Accuracy

The a~~A~~ccuracy is usually measured as a mean~~,~~. ~~it~~ I is calculated as shown below~~,~~ . ~~the~~ The formula of calculating the accuracy is the ratio between the number of correct predictions and the total number of predictions (Mishra, A. 2018). Larger values of accuracy mean that the model is performing well.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}\% \quad (1)$$

4.6.2 Loss Functions

Loss functions are used to determine how many losses occur in the translation model or how the model is far away from the actual data of each cell in the output sequence. This function should be minimized during the training phase~~;~~. Keras offers many types of built-in loss~~es~~ functions such as cross-entropy, mean square error and mean absolute error.

- The cross entropy loss function is used to measure the loss between each predicted probability and the actual class value, which is 0 or 1. ~~small~~

Small differences~~,~~ ~~offers~~ give small scores regarding the logarithmic penalty. This type is usually used in classification problems.

- The m~~M~~ean square error (MSE) is measured by the average of squared differences between the predicted and the actual values~~,~~. ~~t~~The benefit of this function is that ~~it is~~ the effect of large errors becomes more obvious where the model starts focusing on it to minimize it (Golik, *et al*., 2013)~~,~~. ~~t~~The following formula shows the math ~~beyond~~ used in this function.

$$Mean\ Absolute\ Error = \frac{1}{N}\sum_{j=1}^{N}\left(y_j - \hat{y_j}\right)\text{^}2 \qquad (2)$$

- The ~~Mean~~ mean absolute error (MAE) is measured by taking the absolute average value of the differences between the predicted and the actual values as in Equation 3, where N is the total number of predicted samples, y_j is the actual values and $\hat{y_j}$ is the predicted values.

$$Mean\ Absolute\ Error = \frac{1}{N}\sum_{j=1}^{N}\left|y_j - \hat{y_j}\right| \qquad (3)$$

4.6.3 BLEU Score

The Bilingual Evaluation Understudy Score is a metric used to compare the similarity between the two parts ~~for~~ of a translation system~~s~~:. ~~these~~ These parts are the candidate which is the predicted data and the reference which is the actual target data (Papineni *et al*., 2002). This measure is different from accuracy, the bleu calculates the quality of the predicted data for the matching order~~,~~ and the degree of distortion~~,~~. ~~t~~The algorithm calculates the overlaps between the predicted and actual values, where if there are no overlaps in a specific order, then BLEU returns zero. And it is the same metric

used in (Smaıli and Al-Ghawanmeh, 2017) to evaluate their system ~~and we use the same metric to give better results than the reported results in statistical approaches~~.

The BLEU function in Keras is a built in function. It provides sub-functions in evaluation such as, brevity penalty, which is handling the case where the translated sentence is shorter than the actual sentence~~.~~ ~~so i~~It returns low values as a punishment for brevity. Also, it has smoothing functions with seven methods, these methods allow tuning the BLEU score with respect to the problem~~.~~ ~~i~~In our case the best method is the seventh method and it is the same smoothing function used in statistical machine translation model~~.~~ ~~t~~The mechanism of this smoothing function is created to solve the problem of finding the BLEU for short sequence lengths in translated and actual sequences~~.~~ ~~t~~These lengths may have inflated values of BLEU due to having smaller denominators~~.~~ ~~s~~So the author gives them smaller smoothed counts instead of scaling values~~,~~. BLEU is calculated as shown below (Koehn, P. 2009).

$$BLUE = \exp\left(\sum_{n=1}^{N} \log \frac{count\ of\ typical\ matches}{Total\ count\ of\ tranlations}\right) \times BP\% \qquad (4)$$

$$where, BP = brevity\ penalty$$

CHAPTER V

Experiments and Results

Our proposed accompaniment system is based on using encoder and decoder LSTM layers to translate the input vocal data to their corresponding MIDI output value. We used the same corpus used in the statistical machine translation paper. Our expectations are based on the power of the RNN to solve this problem better than the statistical way. The corpus was randomly separated into two sets: training and validation sets. We ~~do~~ did many experiments by Keras ~~TensorFlow~~ with ~~Keras~~ TensorFlow backend to get the best configurations. The next sections ~~will view~~present these experiments and ~~show~~ their results ~~of the best one of them~~.

5.1 RNN Tuning Experiments

Our proposed network architecture is constructed using encoder and decoder LSTM layers~~; we will use the BLEU score as a performance metric of each experimental model,~~. ~~w~~We did several experiments to build the model that ~~recorded~~ gives the highest BLEU result and least loss values. First, we examine the prepared data on the LSTM encoder-decoder model to train the effectiveness of the worked model then we increased the number of layers and tuning the hyper-~~-~~parameters to produce the best translation model.

Tuning the model ~~considered~~ is done by changing the hyper-parameters like batch size, number of neurons, number of layers, ~~the~~ loss, and optimizer functions. The following sections show the change of BLEU score ~~regarding the~~when changing~~e~~ ~~of~~ these hyper-parameters as follows:

5.1.1 Optimizer and Loss Functions

We ~~set up~~used a basic configuration consist~~s~~ing of two layers, 32 ~~number of~~ neurons and 16-sample ~~for the~~ batch size to determine the best loss and optimizer function as shown in Table 6. The best combination is RMSPROP with MAE.

Formatted: Indent: Before: 0"

Table 6: Optimizer and loss experiments

Optimizer	Loss	BLEU/Deg	BLEU/Dur	BLEU/Avg
ADAM	MSE	0.3691	0.3028	0.3359
RMSPROP	MSE	0.3190	0.2859	0.3024
RMSPROP	**MAE**	**0.3856**	**0. 3825**	**0.3840**
ADAM	MAE	0.3563	0.3795	0.3679

5.1.2 Batch Size

We choose root mean square propagation (RMSPROP) optimizer function and MAE loss function from the previous experiments~~.~~. ~~n~~Now we will determine the best batch size in two--layer~~s~~ configuration with keeping the number of neurons equal to 32, as shown in Table 7. The best batch size is 32.

Table 7: Batch size experiments

Batch Size	BLEU/Deg	BLEU/Dur	BLEU/Avg
8	0.3522	0.3790	0.3656
16	0.3856	0. 3825	0.3840
32	**0.4259**	**0.5060**	**0.4660**
64	0.4149	0.4950	0.4549
128	0.4110	0.5090	0.4604
256	0.3613	0.3850	0.3731

5.2 Network Topology

Now we ~~will~~ use the MAE loss, ~~and~~ RMSPROP optimizer function, and the 32-batch size ~~is equal to 32~~ for the next experiments to determine the topology of our model~~,~~. ~~which means~~The objective is to determine the best number of layers and number of neurons as shown in Tables 8, 9 and 10.

Table 8: Topology with one layer experiments

Neurons	BLEU/Deg	BLEU/Dur	BLEU/Avg
8	0.0381	0.0515	0.0448
16	0.2976	0.3355	0.3156
32	0.3851	0. 3181	0.3516
64	0.4002	0. 3652	0.3827
128	**0.3843**	**0.4131**	**0.3987**
256	0.3807	0.4146	0.3977

Table 9: Topology with two layers experiments

Neurons	BLEU/Deg	BLEU/Dur	BLEU/Avg
8	0.3037	0.3969	0.3503
16	0.3642	0.3865	0.3753
32	0.4259	0. 5060	0.4660
64	0.4989	0.4721	0.4855
128	**0.4891**	**0.5880**	**0.5359**
256	0.3784	0.3956	0.3870

Table 10: Topology with three layers experiments

Neurons	BLEU/Deg	BLEU/Dur	BLEU/Avg
8	0.3023	0.3831	0.3427
16	0.3530	0.3781	0.3656
32	0.3411	0.3823	0.3617
64	**0.3556**	**0.3824**	**0.3690**
128	0.3402	0.3808	0.3605
256	0.3521	0.3801	0.3661

As we can see from the tables above, the topology with two layers and 128 ~~numbers of~~ neurons is the best configuration ~~due to the~~with BLEU ~~results,~~score of

53.~~5~~6%, which is considered ~~as~~ a good result compared with the previously published results. Also, the topology with 3 layers does not enhance the results, so there is no need to go ~~in~~ to deeper layers.

5.3 Adding Attention Layer

Our attention function is based on the local attention algorithm~~:~~. ~~w~~We find the attention weights for all hidden states and the sequence length, then the function computes the context vector. In other words, the decoder output should be attended with all previous states and the current sequence length. The results of BLEU score ~~we~~are improved with attention using RMSPROP optimizer function and MAE loss function with changing the number of neurons and layers~~:~~. Table 11 shows the effect of changing neurons number with one layer and attention.

Table 11: Experiments with attention and one layer

Neurons	BLEU/Deg	BLEU/Dur	BLEU/Avg
8	0.2901	0.2187	0.2544
16	0.3841	0.3553	0.3697
32	0.4269	0.3241	0.3755
64	0.3020	0.4758	0.3889
128	**0.4171**	**0. 4029**	**0.4100**
256	0.4134	0.3864	0.3999

Table 12 shows the change in BLEU results while increasing number of layers to two with attention during changing neurons number.

Table 12: Experiments with attention and two layers

Neurons	BLEU/Deg	BLEU/Dur	BLEU/Avg
8	0.3887	0.4140	0.4013
16	0.4244	0.3954	0.4099
32	0.5021	0.4411	0.4716
64	0.3955	0.4150	0.4052
128	**0.6600**	**0. 5518**	**0.6059**

256	0.3901	0.4258	0.4079

Now we test the 3 layers with attention during changing neurons number~~,~~. ~~t~~The results are as shown in Table 13.

Table 13: Experiments with attention and three layers

Neurons	BLEU/Deg	BLEU/Dur	BLEU/Avg
8	0.3602	0.3933	0.3767
16	0.4212	0.3949	0.4080
32	0.4180	0.4052	0.4116
64	**0.4251**	**0.4881**	**0.4566**
128	0.3633	0.3830	0.3731
256	0.3562	0.4322	0.3942

The results show that attention mechanism ~~is~~ improve~~d~~ the BLEU score in all layers~~, the effect of improvement will be noticeable when the corpus expands~~. Figure 11 shows the comparison between the previous ~~tables~~best results.

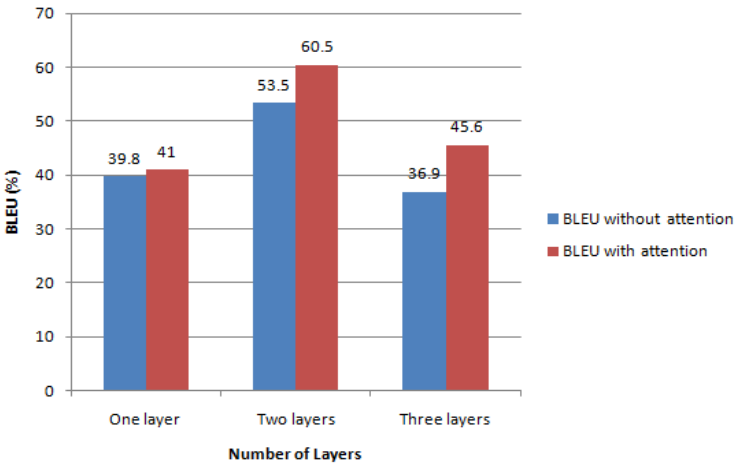


Figure 11: Best results with and without attention

5.4 Training Convergence

The used loss function in our work is MAE. It is performing ~~the best values with~~well in all experiments ~~which are less than~~below 25%. In ~~O~~our proposed model ~~which~~ that reached the highest BLEU score~~s~~ with 60.~~5~~6%, the loss function was ~~the least value upon others with~~minimum at 0.183~~18.3% with attention~~.

Choosing the number of epochs is a critical ~~point~~decision, where increasing the number of epochs cause~~s~~ ~~O~~overfitting the training datasets~~.~~. ~~w~~Whereas few numbers of epochs may cause under fitting the model~~.~~. ~~I~~In our model, we used early stopping function which helps to determine the accurate number of epochs where the model should stop training. ~~I~~In other words, we can define early stopping function as a monitor ~~where~~ which looks for the point in which the model performance stops improving~~.~~. ~~I~~In our case, we set the monitor to watch out the variation of the validation losses with setting the patience to five~~.~~; ~~that means~~ after five epochs if the model validation loss does not enhance, stop training~~.~~. Figure 12 shows the accuracy values when we used two layers with 32 batch size and 128 ~~numbers of~~ neurons without attention layer.

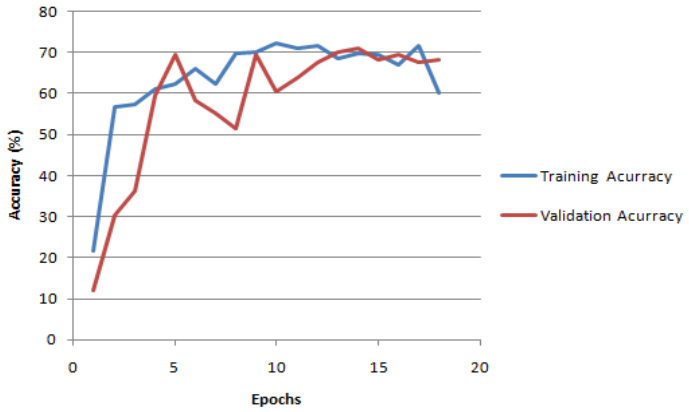


Figure 12: Accuracy values for the best model without attention

The accuracy values are not stable in our experiments because the data is not sequenced according to a particular pattern it is changing between positives and negatives values suddenly~~.~~ ~~s~~So we cannot use it as a monitor of early stopping function. Figure 13 shows the losses of the best model, where it has two layers and 32 batch sizes with 128 ~~numbers of~~ neurons without attention.

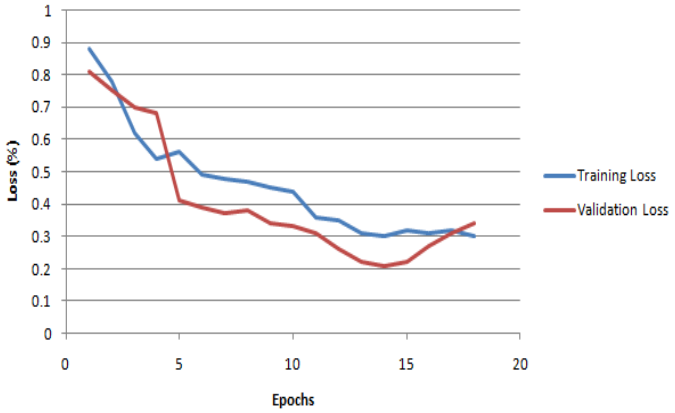


Figure 13: Loss values for the best model without attention

Now, we should test the effect of attention layer on accuracy and loss values~~.~~ ~~t~~The accuracy and loss values ~~are~~ improve~~d~~ with attention layer as shown in Figure~~s~~ 14 and ~~Figure~~ 15.

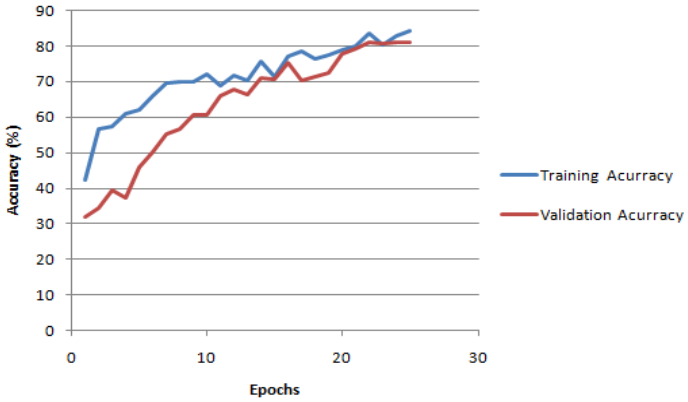


Figure14: Accuracy values for the best model with attention

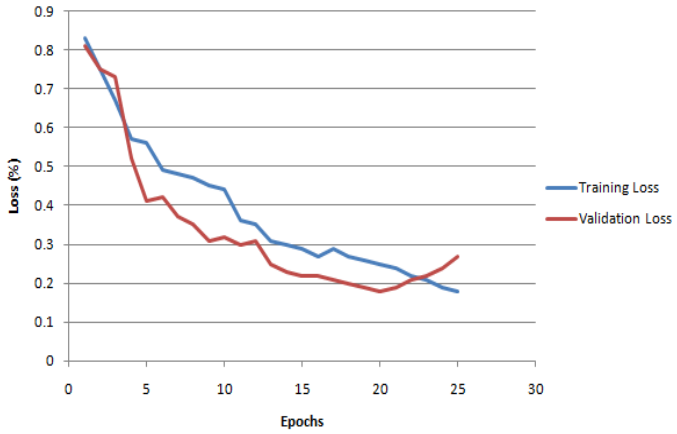


Figure15: Loss values for the best model with attention

The ~~A~~attention layer ~~has~~ improve~~d~~ the performance metrics and the model does not reach~~ed the O~~overfitting in earlier epochs~~,~~; more epochs, means more training.

5.5 Training Time

The elapsed time in machine translation is important to measure~~,~~. ~~t~~The training time was long ~~and this time should be reliable to work on~~which is expected when

working with large data sets. Also, the testing time is useful to know how much time we need to translate the sequences.

Training and testing times ~~effects~~ are affected by the number of layers and the number of neurons, because more layers mean more weights to train which needs more time ~~and the same for testing time~~. Also, increasing the neuron numbers increase~~d~~s the complexity of the comparison between data representation in the model which makes it need more time to train and test.

From previous results we do not need more than two layers ~~to train and test the data~~ and neuron~~s~~ number was changed within ~~four~~ six values (8, 16, 32, 64, 128 and 256), Figure 16 shows the increasing of time when we increase the number of neurons using one layer and Figure 17 shows the change in BLEU results regarding the change in neuron numbers.

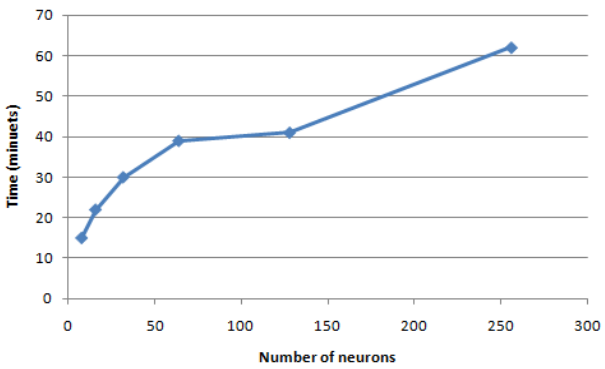


Figure 16: Timing when increasing the number of neurons for 1 layer

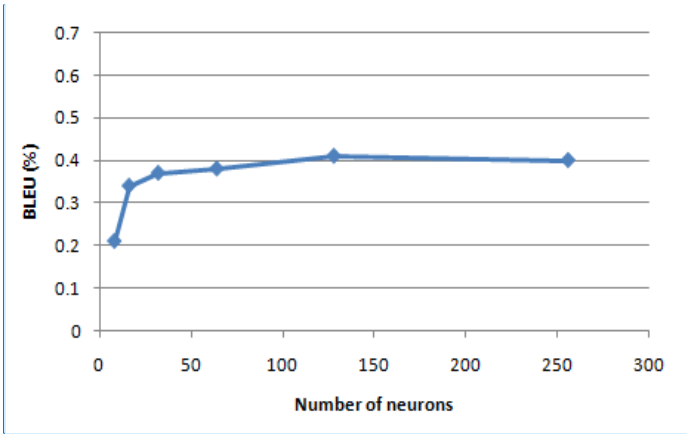


Figure 17: BLEU scores when increasing the number of neurons for 1 layer

Now we test the time and BLEU for two layers with the same neurons variation number ~~which~~ as shown in Figures 18 and ~~Figure~~ 19.

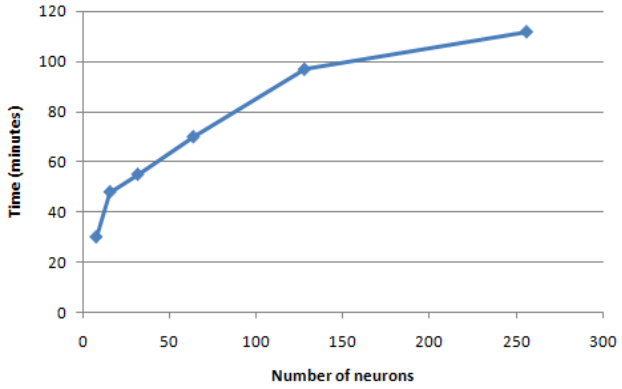
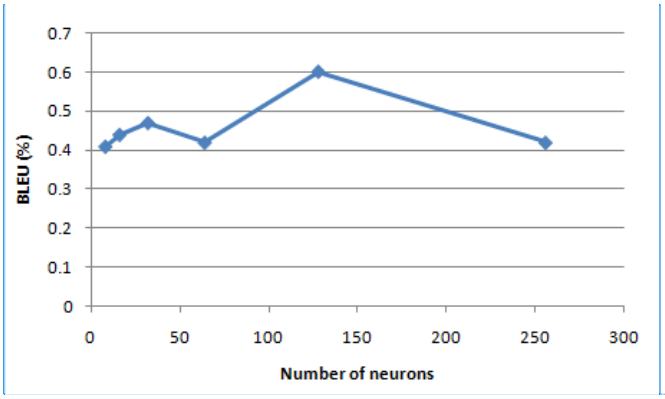


Figure 18: Timing when increasing the number of neurons for 2 layers

Commented [GA10]: Remove (%) from the vertical axis. Also from Figures 13 and 15



Commented [GA11]: Remove (%) from here also

Figure 19: BLEU scores when increasing the number of neurons for 2 layers

5.6 Discussion

After all the experiments applied using RNN we found that encoder-decoder RNN with two layers using RMSPROP and MAE as an optimizer and loss functions with 128 neurons ~~numbers~~ and 32 batch size achieves the best results~~ among the other experiments~~. Also, the results are better than the state-of-art approaches of automatic accompaniment fields.

Adding attention to the model improved the results because it is concentrated the decoder in both training and prediction phases to the important parts of features in the input sequence.

Figure 20 shows the comparison between our results in translation the data in both sides from vocal to instrumental, and from instrumental to vocal data, using Machine Learning Translation (MLT), and the Statistical Machine Translation (SMT) results ~~in both sides which published by~~ (Smaıli and Al-Ghawanmeh, 2017)~~ paper~~.

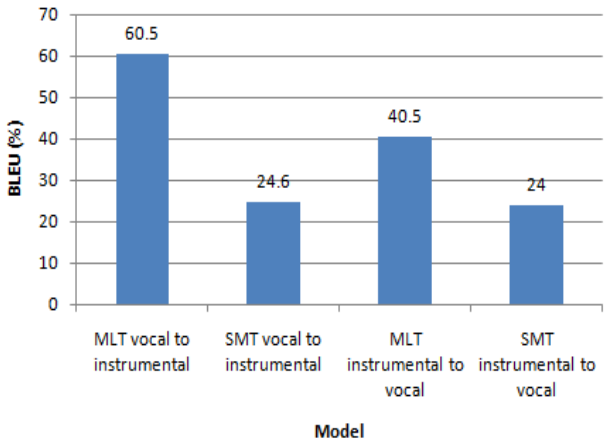


Figure 20: Comparison between MLT and SMT models

CHAPTER VI

Conclusions and Future Work

The process of automatically translate~~e~~ing Arab vocal improvisation sentences to the instrumental sentences is called ~~an~~ automatic accompaniment system. It relies on inserting vocal input data generated by singers and find~~s~~ing ~~the correct relationship with its~~ corresponding MIDI data or instrumental data. We used ~~as an output using~~ the BLEU

score to determine the ~~correctness between~~similarity of the resulting data that predicted by the system and the ~~actual~~ target data ~~that stored in the transcriptions~~.

The techniques used in solving the accompaniment problem are based on RNN with encoder and decoder LSTM architecture that receive~~d~~ vocal transcription recorded manually and the output is instrumental transcriptions that were generated also manually by an instrumentalist in closed rooms to prevent any noise that could affect the data and results (Al-Ghawanmeh and Smaïli, 2017).

To improve the BLEU score for the proposed accompaniment system, we used attention mechanism that focuses on the sequence lengths of the translated and target sets. Also it is informing the decoder to pay attention on the first note which is called the key note. This note is determined the method of listing the next notes in the same sequence note.

In this work, we use encoder-decoder LSTM architecture to train the data corpus ~~which~~ consist~~s~~ing of 1933 vocal sentences and their corresponding MIDI sentences, and then we reconstruct the encoder-decoder model to do inference on the validation set which consists of 829 parallel vocal and MIDI sentences. We used the BLEU score, MAE loss function, and accuracy to measure the performance of our model~~.~~. ~~t~~The overall accuracy in our model reached 71.2%, the loss was 0.20~~.~~1~~%~~ and 53.~~5~~6% for BLEU score. When we used attention layer the accuracy reached 82% and the loss was 0.1~~8~~% and 60.~~5~~6% BLEU score with two layers and 128 neurons and this is the best results over all experiments that were applied on the same corpus.

We used data pre-processing techniques to remove the noisy data that affects the performance of the model. Also, we used the masking layer to deal with a variable

sequence length of the input. The attention layer added significant improvement in the results and improved ~~the~~ all performance metrics. In the future, we can use other methods of attention mechanism to gain higher BLEU results. Also, we can extend the transcriptions to train the model on more data sets.

Our research on the computer processing of Arabic music is important because most of the researchers wrote about non-Arabic music as in (Toyama, M. 2001) and all published papers which handle Arabic accompaniment system~~,~~ solved the problem using statistical machine translation and classical machine translation. Many commercial applications of the Arabic automatic accompaniment system were published based on the results generated with the statistical approaches model. We appreciate their work~~s~~ but we are confident that machine learning will give better results if converted to application in the future.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Kudlur, M. (2016). TensorFlow: A system for large-scale machine learning. **In 12th ~~{~~USENIX~~}~~ Symposium on Operating Systems Design and Implementation** (~~{~~OSDI~~}~~ 16), pp. 265-283.

Al-Ghawanmeh, F. and Smaïli, K. (2017), Statistical Machine Translation from Arab Vocal Improvisation to Instrumental Melodic Accompaniment. Proceedings of **International Conference on Natural Language, Signal and Speech Processing ICNLSSP**.

Formatted: Font: Bold, Complex Script Font: Bold

Al-Ghawanmeh, F. M., Al-Ghawanmeh, M. T., & Abed, M. W. (2016). Real-Time Maqam Estimation Model in Max/MSP Configured for the Nāy. **International Journal of Communications, Network and System Sciences**, 9(02), 39.

Al-Ghawanmeh, F. M., & Shannak, Z. R. (2015). Automatic Accompaniment to Arab Vocal Improvisation: From Technical to Commercial Perspectives. **Journal of Software Engineering and Applications**, 8(01), 16.

Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. **arXiv** preprint arXiv:1409.0473.

Burges, C. J. (1998), A Tutorial on Support Vector Machines for Pattern Classification, **Data mining and Knowledge Discovery**, 2(2), pp.121-167.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder-decoder for statistical machine translation. **arXiv** preprint arXiv:1406.

Chollet, F. (2016). Building autoencoders in ~~keras~~Keras. **The Keras Blog**.

Dannenberg, R.B. (1984), An on-line algorithm for real-time accompaniment. Proceedings of the **International Cryptographic Module Conference (ICMC)**, October, 84, pp. 193-198.

de Oliveira, H. M., & de Oliveira, R. C. (2017). Understanding MIDI: A Painless Tutorial on Midi Format. **arXiv** preprint arXiv:1705.05322.

Gao, J., & He, X. (2013). Training MRF-based phrase translation models using gradient ascent. Proceedings of the **2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**, pp. 450–459.

Géron, A. (2017). Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems. "O'Reilly Media, Inc.".

Golik, P., Doetsch, P., & Ney, H. (2013, August). Cross-entropy vs. squared error training: a theoretical and experimental comparison. In Interspeech, 13, pp. 1756-1760.

Graves, A. Jaitly, N and Mohamed, A. (2013), hybrid speech recognition with deep bidirectional LSTM, University of Toronto, Canada.

Graves, A. and Schmidhuber, J. (2005), Framewise phoneme classification with bidirectional LSTM and other neural network architectures, **Neural Networks,** 18(5-6), pp. 602-610.

Gulli, A. and Pal, S. (2017). Deep Learning with Keras. **Packt Publishing Ltd**.

Gulli, A. Pal, S. (2017), implement neural networks with ~~keras~~ Keras and Theano and Tensoflow, **Packt Publishing Ltd**, pp. 60_93.

Haddad, R., Al-Ghawanmeh, F.M. and Al-Ghawanmeh, M.T. (2010), Educational tools based on MIR system for Arabian woodwinds. **Journal of Music, Technology & Education**, 3(1), pp. 31-46.

Hochreiter, S. Schmidhuber, J. (1997), Long short-term memory, **Neural computation**, 9(8), pp. 1735-1780.

Kaiming, H. Xiangyu, Z. Shaoqing, R. and Jian, S. (2016), Identity mappings in deep residual networks, **In European conference on computer vision**, pp. 630–645.

Ketkar, N. (2017). Introduction to ~~keras~~Keras, **in Deep Learning with Python.** pp. 97-111. Apress, Berkeley, CA.

Koehn, P. (2009). Statistical machine translation, Cambridge University Press, pp. 17–20.

Larson, S. (2005). Composition versus Improvisation, **Journal of Music Theory**, 49(2), pp. 241-275.

Luong, M. T., Pham, H., & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. **arXiv** preprint arXiv:1508.04025.

Mishra, A. (2018). Metrics to evaluate your machine learning algorithm. Towards Data Science. URL: https://towardsdatascience.com/metrics-toevaluate-your-machine-learning-algorithm-f10ba6e38234. Accessed, 12, 2018.

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002, July). BLEU: a method for automatic evaluation of machine translation. Proceedings of the **40th annual meeting on association for computational linguistics**. pp. 311-318. Association for Computational Linguistics.

Simon, I., Morris, D., & Basu, S. (2008, April). MySong: automatic accompaniment generation for vocal melodies. Proceedings of *the* **SIGCHI conference on human factors in computing systems**. pp. 725-734. ACM.

Svozil, D., Kvasnicka, V., and Pospichal, J. (1997). Introduction to multi-layer feed-forward neural networks. **Chemometrics and intelligent laboratory systems**, **39(1)**, pp. 43-62.

Tan, P. N., Steinbach, M., and Kumar, V. (2006**).** Classification: basic concepts, decision trees, and model evaluation. ~~**data**~~**Data mining,** pp. 145-205. Boston: Pearson Addison Wesley.

Toyama, M. (2001). Song accompaniment system, **U.S. Patent** pp. 6-153. Washington, DC: U.S. Patent and Trademark Office.

Formatted: Font: Bold, Complex Script Font: Bold

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need, Proceedings of the **neural information processing systems**, pp. 5998-6008.

Vinyals, O., Kaiser, L., Slav Petrov, T., Sutskever, Hinton, E. (2014) Grammar as a foreign language.**The Computing ResearchRepository**, abs/1412.7449.

Wilkins, Dennis (2017). "PG Music Band In A Box 2017", Proceedings of the **Sound Magazine**.

Zhang, J., and Zong, C. (2015). Deep neural networks in machine translation: proceedings of the **IEEE Computer Society**, pp. 1541-1672.

الترجمة من الإرتجال الصوتي إلى المرافقة اللحنية بإستخدام لغة الآلة

إعداد
ايه محمود محمد الشمايله

المشرف
الدكتور غيث علي عبندة

ملخــص

الإرتجال الصوتي هو من أهم أنواع الموسيقى العربية وأقدمها, في هذا النوع الموسيقي يتم سرد أبيات شعرية يتخللها مرافقة لحنية تنسجم مع طبيعة السرد حيث يقوم العازف بإدراج جمل موسيقية عندما يتوقف المغني عن السرد و هذا النوع من الموسيقى يسمى الموال في شرق آسيا ويدعى إستخبار في بلاد المغرب العربي.

جميع الأبحاث المتخصصة في هذا المجال تسلط الضوء على طريقة المرافقة اللحنية للإرتجال الصوتي في الموسيقى الغربية وقليل منها ما يركز على الموسيقى الشرقيه لذلك وجب النظر في إستحدام الطرق المحوسبه لتسخيرها في خدمة الموسيقى العربية من أجل تطويرها وتسهيل إستخدامها من قبل الموسيقيين وغير الموسيقيين.

الطرق المستخدمة في مجال المرافقة اللحنية للموسيى العربية قليلة جداً, ومن أهم الطرق هي الترجمه الآلية الإحصائية, حيث يستخدم الباحث طرق إحصائيه لإيجاد توافق لحني ما بين الإرتجال والجمل الموسيقية من خلال إستخدام الترجمة الآلية وهي طريقة تقليدية تعتمد على تغيير أحد المتغيرات الرئيسيه التي تشكل هذه الطريقة, تتلخص المتغيرات الثلاثة في: المتغير الذي يعتمد على طريقة تقديم الجمل الموسيسقية, المتغير الذي يقلل الجمل الموسيقية ومتغير دمج الجمل الموسيقية, النظام المعتمد في هذا البحث حصل على أفضل النتائج في هذا المجال بإستخدام مقياس التقييم الثنائي اللغه المستخدم لمعرفة دقة الترجمة من جملة إلى جمل وقد حصل على نسبة 24.62% في مرافقة الإرتجال الصوتي إلى الجمل الموسيقية المناسبة وعلى 24.07% في الإتجاة المعاكس وهي أعلى نتيجة أحرزت حتى يومنا هذا.

في هذا العمل تم إقتراح طريقه لحل مشكله مرافقة الجمل الصوتيه باللحن المناسب بإستخدام الترجمه بواسطه الشبكات العصبونيه المتكرره ولتحسين النتائج قمنا بتعديل المدخلات والمخرجات للنظام لتلائم عمليه الترجمه بإستخدام الآلة.

حقق النظام الذي تم إقتراحه في هذا العمل نتائج أفضل من النتائج المنشورة مسبقآ في هذا المجال فقد أحرز نظامنا المقترح على نسبه 60.5% في مرافقه الإرتجال الصوتي إلى الجمل الموسيقيه بإستخدام مقياس التقييم الثنائي الذي أستخدم بالأبحاث السابقه وحصل على 40.5% في التحويل المعاكس من الجمل الموسيقيه للجمل الصوتيه وهذة أفضل نتيجة قد أحرزت إلى الآن في هذا المجال.