Automatic Composition of Lyrics and Music for a Song in Macedonian using Deep Learning

Angela Najdoska, Emilija Kotevska, Tamara Markachevikj, Hristijan Gjoreski
Faculty of Electrical Engineering and Information Technologies
Ss. Cyril and Methodius University in Skopje, Macedonia
angela.najdoska17@gmail.com, emilijakotevska@gmail.com, markacevic@gmail.com

Abstract **- This paper presents an approach to develop deep learning models to automatically generate music and lyrics for a song in Macedonian language. The approach consists of two steps: (i) a deep learning model to generate text (lyrics) for songs using Gated Recurrent Unit approach, and (ii) a deep learning model for music generation for the lyrics of the song generated in the previous step. The model for lyrics generation is based on Gated Recurrent Units algorithm, and the model for music generation is based on a pre-trained Recurrent Neural Network and Variational Autoencoder model, which generates sequences of music that are later combined to produce the final music for the song. The music generation model interpolates between two-note sequences and thus completes the process of creating the music. The results show that our model generates understandable lyrics for songs and decent quality music for the same. The result songs are unique, new, and after some melodical post-processing, can be potentially used to help the process of song creation**.

Keywords— Text generation; Lyrics generation; Macedonian song; GRU; Deep learning; RNN; Machine learning.

I. INTRODUCTION

Music is the art of arranging sounds in order to produce a composition throughout the elements of melody, harmony and rhythm. General definitions of music include common elements such as pitch, rhythm, dynamics, timbre and texture. In many cultures, music is an important part of people's way of life, as it plays a key role in social and cultural activities. The music industry includes the individuals who create new songs and musical pieces, individuals who perform and record music, who organize concert tours and who sell recordings, sheet music and scores to customers.

Machine Learning, as an important segment of Artificial Intelligence, is increasingly present in our daily lives. Text generation with the help of Machine Learning and Deep Learning is popular in every industry, especially in mobile phones, applications, data-science and even journalism [1]. Every day, we come across text generation: iMessage autocomplete, Google search, Gmail smart typing are just a few examples. Using automatic text generation, we can make many processes easier, for example: composing a text (lyrics) for a song and producing a music for the same. Writing a song and a melody for the same is a challenging process that requires inspiration and artistic way of thinking. Having a system that can automatically produce several dozens of lyrics and music for the same, can be if great importance and help. Even if the composer and the music producer have to manually go through them, and analyze and; these artificially generated songs can still be used as a basis and inspiration and this way making the whole process easier.

In this paper we propose a Deep learning approach to generating lyrics for songs in Macedonian language. Additionally, we propose a Deep Learning approach for composing music for the same songs. Even though, there have been approaches for generating poems, news articles in Macedonian language [2] to the best of our knowledge, this is a first study that tackles the problem of generating songs and music in Macedonian language using Deep Learning.

II. DATASET AND PREPROCESSING

For the purposes of this study, we created our own dataset - because there is no publicly available repository and database for the same. We used the lyrics of almost all the songs performed by one of the most famous Macedonian singers - Toshe Proeski. We used 64 songs, each with a different number of verses, and a different metric. The total number of words in the database is 10731. Next, we performed data preprocessing and preparation for the model. This included, parsing, tokenization and equalizing the number of tokens in each verse. Then, we removed the punctuation marks and converted all letters to lowercase. Then, we transformed the corpus so that each verse of the database represents one element of the corpus. Finally, we performed padding, so that all the sequences have the same length. The data and the code are available online [3].

III. LYRICS GENERATION MODEL

To generate the lyrics of a song, we used a sequential type of Deep Learning approach, in particular Recurrent Neural Network (RNN). The model is shown in Figure 1, and consists of:

- Embedding Layer with number of inputs equal to the number of different words in the corpus (the number of tokens in sequence decreased by one) and 5 outputs.

- Two hidden layers of Gated Recurrent Unit (GRU) [4] which is an RNN variant that uses multiplicative connections which allow the current input character to determine the transition matrix from one hidden state vector to the next.
- Dropout layer is used to reduce overfitting [5]. This technique works by randomly dropping units (along with their connections) from the neural network during training. This prevents units from co-adapting too much. The number of neurons in both layers is 80 and 120 respectively.
- The last layer is a Dense layer which has outputs equal to the number of different words in the corpus and “softmax” activation function.

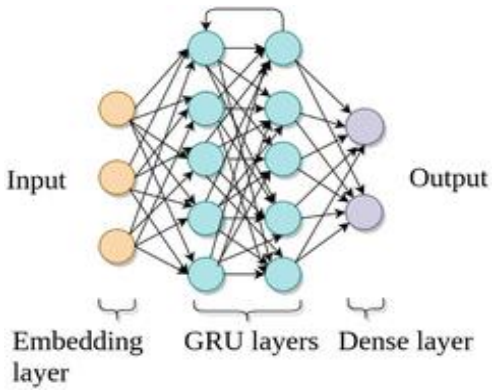

Figure 1: Model architecture: input, embedding layer, GRU layers, Dense layer and output.

We used Categorical cross-entropy as a cost function and optimize it with Adam optimizer (Adaptive Moment Estimation), which is method that computes adaptive learning rates for each parameter [6]. It is an algorithm for first-order gradient-based optimization of stochastic objective functions. For monitoring the accuracy of the model, in the training process, we use accuracy metrics.

We trained the model for 400 epochs and used EarlyStop callback in order to prevent overfitting. From the existing corpus, we make a new corpus whose elements are words, in order to choose a random word that we will use as a seed on which the whole song will be generated.

Finally, the text for the song was generated verse by verse. The first generated word use the one-to-one principal, while all subsequent words in the verse are generated by many-to-one principal. Moreover, the last word from the previous verse is the seed for the next verse. Arbitrarily, we chose the song to consist of 4 stanzas with 4 verses each, and each verse to contain 6 words.

IV. MUSIC GENERATION MODEL

After the lyrics was generated, the next step was to generate a music for the same. For music generation, we used the Magenta open-source Python library [7]. Despite Magenta, we also used the node_seg library which allows abstract representation of a series of notes, each with different pitches, instruments and strike velocities, much like the MIDI standard (Musical Instruments Digital Interface).

In the Magenta Library we used a pre-trained Melody RNN model - a model of the Recurrent Neural Network for generating a sequence of notes based on an initially given sequence. We used this model to generate two different music sequences. First, we created the initial sequences that are actually part of two songs by ToshеProeski: "Ledena" and "Po tebe". This is shown in Figure 2. Then, in the Melody RNN model we set certain parameters such as temperature (degree of randomness of note generation), number of steps (the length of the generated sequence depends on it), and tempo. This is shown in Figure 3.

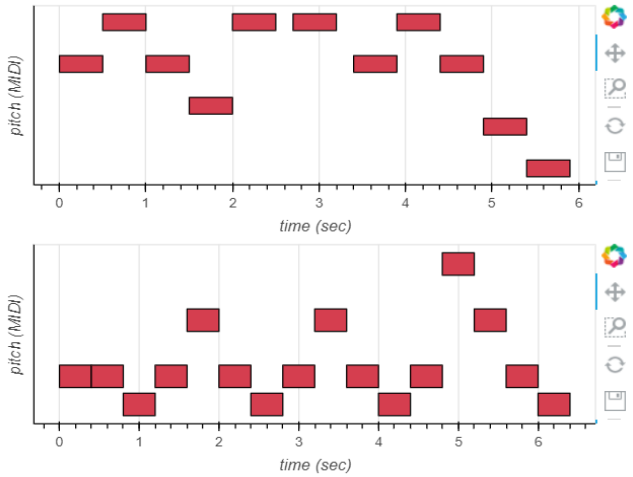

Figure 2: Initial sequences-parts of Toshe Proeski’s songs.

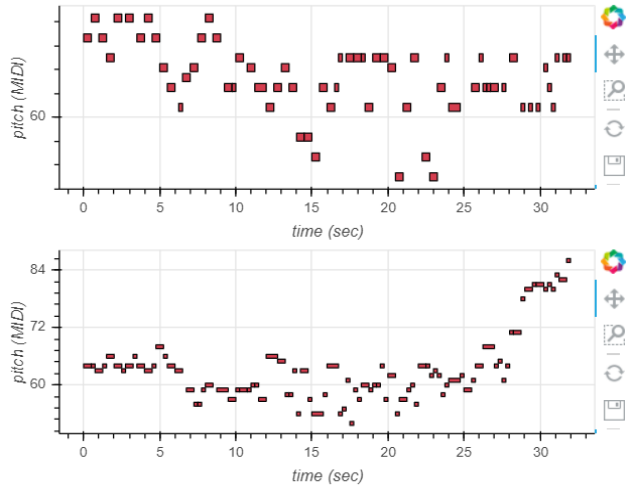

Figure 3: Sequences generated by Melody RNN.

In the next step we used the Magenta Studio, which is a collection of applications built on Magenta tools and models. This collection contains the applications: Continue, Groove, Generate, Drumify and Interpolate, which allow implementation of the models from the Magenta library on midi files. Interpolate is an application that takes two drumbeats or two tunes as inputs. It can create up to 16 files that combine the qualities of the two input files. It is useful

for merging musical ideas or for creating a smooth match between them.

We used Magenta Studio to merge the two previously generated musical sequences. This application is based on the Music VAE (Variational Autoencoder) model which has a hierarchical recurrent variational autoencoder for learning latent spaces for musical scores. Latent spaces are capable of learning the fundamental characteristics of a training dataset and they are able to represent the variation of real data in a lower-dimensional space [8]. Autoencoders are different from Recurrent Neural Networks since they use an encoder and a decoder to learn from the data. In the case of music, a sequence of music notes is encoded into a latent factor and then decoded into a sequence again [9]. One way of thinking about VAE is like mapping from MIDI to compressed space in which similar music patterns come together. Each of your input forms is represented by a position on this map. The interpolation draws a line between these two positions and returns clips along this line. A visual representation of the Music VAE model is shown in Figure 4.

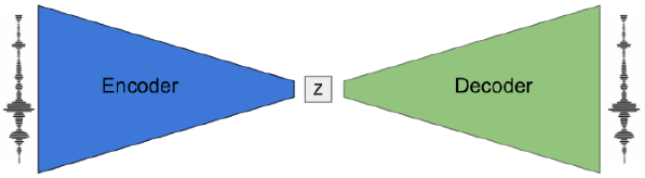

Figure 4: Structure-part of the Music VAE model

The number of recovered files is set with the steps slider. Each of the applications has a temperature slider. Temperature is a parameter used for taking samples in the last layer of the neural network. This parameter is used to generate randomness. Higher values create more variation, and sometimes even chaos, while lower values are more conservative in their predictions.

V. Experimental Results

In our experiments, first we generated several lyrics for a particular song using the lyrics generation model. After that, we qualitatively inspected them and chose the most interesting one. In the next step, after applying interpolation on the two previously generated sequences, we got the music for the song. We attached the two melodic sequences, selected a temperature equal to 1.5 and generated 5 files, by setting the steps parameter. Then, we merged these files into one, with the help of an online converter for combining MIDI files into one midi file, in order to get a longer music file [10]. This way, we got music with a duration of 32 seconds. The result song is shown in Figure 5. The music for the same can be accessed online [3], where we also included most of the other generated songs.

The lyrics of the song (see Figure 5) is understandable, and the ones that know Toshe's songs can recognize some of the style and the phrases which are also used in his previous songs. However, the whole text is new, has new phrases and combination of words, which makes the song unique and after some post-processing, can be further used.

во окото сјајот не поминува не
не е месечината во ноќва што
ме привлекува што во глава ти
фали си слатка но и студена

ми илјада и двеста прегратки насмевка
и ледена и медена и да
ми илјада и двеста прегратки насмевка
и ледена и медена и да

усни на усни да те води
познат и двеста дена суви со
чуда по тебе изгорев јас и
каде ли траг на вратот криеш

дали е доцна нешто да сменам
да те најдам ради нас патуваш
пак не е грубо ако е
од тебе во него на ум

Figure 5: The lyrics for the generated song.

In our experiments, the part with the automatic text (lyrics) generation was the most challenging. The music generation process was more of a usage of already existing tools and models. For the lyrics generation we tried different types of Deep Learning approaches and neural networks architectures, various number of hidden layers, various number of neurons, and also different activation functions. Here we explain the process, and the results achieved by the different combinations.

Firstly, we tried the Long Short Term Memory Neural Network (LSTM) approach [11], which consisted of an Embedding Layer with a number of inputs equal to the number of different words in the body, 7 outputs and input_length equal to the number of tokens in sequence decreased by one. However, the results that we got with the GRU approach were better for several percentage points compared to the LSTM, therefore in the next optimization steps we used GRU.

Next, we tried several combinations of adding hidden layers. The experiments showed us that increasing the number of hidden layers, leads to a decrease of accuracy. We speculate that the reason for this is overfitting, i.e., additional layers contribute additional parameters and more complex model, which for a small dataset overfits more easily. Therefore, we fixed the number of hidden layers at two, which provided the best accuracy in the experiments.

Next, we tried two activation functions: RELU and softmax. The results showed us that the model with the RELU function achieves significantly lower accuracy compared to the softmax [12].

Finally, we tried reducing the number of outputs from the Embedding layer, and the highest accuracy was obtained with 5 outputs - this was used in the final model. The final accuracy of the model was 72% - which is in line with

similar tasks in the literature [2][11]. Please note, that the accuracy in this task is calculated as a ratio of the correctly predicted words from the total number of words.

VI. Conclusion

The paper described our research in the area of automatic song lyrics generation in Macedonian language and producing a music for the generated songs. In the process, we were able to develop two Deep Learning models for the two tasks: lyrics generation and music generation. The models were developed and tested using all the songs sang by one of the most famous Macedonian singers - Toshe Proeski. The results showed that the generated songs are unique, new, and after some post-processing can be potentially used to help the process of song creation.

The dataset, the models and the code are available for usage [3]. In this format, the models can be used by people that have basic understanding of Python and Machine Learning. However, in the future we plan to release a version with Graphical User Interface, so that music practitioners can freely use the software without prior knowledge of coding.

We envision that our models will be used as a tool that will help songwriters and producers in the process. In the future, the model could be further enhanced by adding more songs from other Macedonian performers to the database and by generating sounds from another instrument, such as drums.

To conclude, AI is very useful for song generation, but it cannot completely replace human resources. However, it is good enough to help in the process, and it has a predisposition to improve over time.

References

[1] Iqbal, Touseef, and Shaima Qureshi. "The survey: Text generation models in deep learning." Journal of King Saud University-Computer and Information Sciences (2020).

[2] Ivona Milanova, Ksenija Sarvanoska, Viktor Srbinoski, Hristijan Gjoreski. Automatic Text Generation in Macedonian Using Recurrent Neural Networks. ICT Innovations 2019

[3] The code and the dataset for this research. Online, accessed July 2021: https://github.com/ekotevska/Macedonian_song

[4] Gao, Yuan & Glowacka, Dorota. Deep Gate Recurrent Neural Network. Proceedings of The 8th Asian Conference on Machine Learning, PMLR 63:350-365 (2016).

[5] Nitish Srivastava and Geoffrey Hinton and Alex Krizhevsky and Ilya Sutskever and Ruslan Salakhutdinov.. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 1929-1958. (2014)

[6] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[7] Magenta library. Online, accessed July 2021: https://magenta.tensorflow.org/

[8] MusicVAE: Creating a palette for musical scores with machine learning. Online, accessed July 2021: https://magenta.tensorflow.org/music-vae

[9] Tikhonov, Alexey, and Ivan P. Yamshchikov. "Music generation with variational recurrent autoencoder supported by history." arXiv preprint arXiv:1705.05458 (2017).

[10] MIDI file processing. Online, accessed July 2021: https://www.ofoct.com/merge-midi-files

[11] Pawade, D., et al. "Story scrambler-automatic text generation using word level RNN-LSTM." International Journal of Information Technology and Computer Science (IJITCS) 10.6 (2018): 44-53.

[12] Steffen Eger, Paul Youssef, Iryna Gurevych. Comparing Deep Learning Activation Functions across NLP tasks