International Journal of User-System Interaction 13(2) 2020, 61-72

Text generation and music composition by human and neural network interaction

Răzvan Păroiu[1], Ștefan Trăuşan-Matu[1,2,3]

[1]University POLITEHNICA of Bucharest
Splaiul Independenței 313, București 060042
[2]Research Institute for Artificial Intelligence
[3]Academy of Romanian Scientists
E-mail: razvan.paroiu@gmail.com, stefan.trausan@upb.ro

Abstract. Artificial intelligence has been used for a long time to generate natural language text or music, artifacts that are usually considered as the creation of human minds. However, text and music generated with artificial intelligence in general lack the feeling specific to human creations. Therefore, hybrid approaches where the user generates content by interacting with the artificial intelligence have provided better results. This paper presents an application that follows this hybrid approach, which can be used by a human to create text and compose music with the help of artificial intelligence. Text or musical notes are generated by multiple trained neural network models and the user is the one that selects the best-generated result.

Keywords: artificial creativity, natural language processing, language models, music composition

DOI: 10.37789/ijusi.2020.13.2.1

1. Introduction

According to Mihaly Csikszentmihalyi (1996), human creativity is influenced not only by genetic inheritance but also by the domain in which is used and the group of peoples that represents the domain and judge the value of the new creations. This point of view defines multiple areas where artificial intelligence can make improvements on human creativity.

When the creative process takes place, Csikszentmihalyi remarked that creative persons enter into a flow state, and that multiple factors have been recognized to be crucial for entering in this state of mind Csikszentmihalyi, 1996). Some of these factors might be enhanced by artificial intelligence. Lateral thinking improvement techniques (Bono, 2009) are also known to improve creativity and curiosity, and artificial intelligence has already been

used to improve the usage of these techniques (Oprisan & Trausan-Matu, 2013).

On the other side, artificial intelligence has also progressed enough that for some well-defined tasks that do not imply human subjectivity, such as the fidelity of imitating an artist, intelligent algorithms have overtaken even the human experts[1].

In the case of art generated by artificial creativity, that means created using artificial intelligence, human intervention is still important, because it is not very clear how to develop algorithms that generate text or music that create complex human emotions, similarly to a real writer or composer. For example, starting from the evolutionist ideas, for improving genetic algorithms, human intervention has been used for a long time to replace the decision process of the fitness function, especially in areas such as music composition where mathematical formulas that scored the quality of the chromosomes that represent parts of music, do not offer good results (Bruce, 1995).

In this paper it is proposed an approach of human creativity enhancement with the help of artificial intelligence. The approach used for the purpose of content generation is based on neural network language models. A predicted content is generated based on a user created input and it represents the continuation sugestion of the neural network for the respective input. The sugestions are then displayed so that the user could visualize and select if some parts of them he wants to use in his own created content.

2. State of the art

Creativity is the ability of a person to create something new such as a piece of art or an idea. Creativity stimulation techniques include group methods such as brainstorming, synectics, Delphi, General Morphological Analysis, or single person-oriented methods such as lateral thinking techniques (Bono, 2009). Current group methods for creativity stimulation have been improved by the use of artificial intelligence (Oprisan & Trausan-Matu, 2013).

Creativity stimulation is best obtained when the application is focused on one specific domain. A lot of research has been made where music is

[1] The Next Rembrandt [Video file]. Retrieved at 6.07.2020 from https://www.youtube.com/watch?time_continue=37&v=IuygOYZ1Ngo&feature=emb_logo.

generated by the interaction between the user and the machine. Magenta (2020) is such an example. It is an open source tool used for digital music composition build in TensorFlow and it is used to generate MIDI files by using neural networks. The platform is extensible, plugins have been built on top of it and some of them are created by an open source community. The majority of plugins use some sort of user input, which are used for generating music. Some examples of plugins are: A.I. Jam, where a duet is made between the notes played by the user and the notes generated by the machine learning algorithms (Kingma & Lei Ba, 2014), Sornting, where the user draws sequences of notes and the algorithm generates intermediate sequences (Adam et al., 2019), or NSynth Sound Maker, which generates new sounds by mixing two different instruments Jesse et al., 2017).

3. Application description

The application described in this paper, implements multiple language models that can sugest text or music after the user has typed some input words or musical notes in a dedicated input area. The suggested generated text has the length of 50 words and the suggested music has the length of 20 notes. The user not only can select the preferred sugested output, but she can also select how much of the generated content should be appended to her written content by hovering with the cursor over the word that ends the wanted selection (for example, the word “gentleman” in Figure 1) and the application automatically marks with red all the selection, thus enhancing the interactivity of the application. When the user clicks on the word, the whole text marked with red is inserted into the input area and the language models will make a new sugestion based on this modification using asynchronous AJAX requests, further increasing the interactivity of the application. The user can also modify at any moment the content in the input area and the language models will make another new sugestion.

A similar process is provided by the application for music composition. The sound interpretation of the musical notes that the user enters on the interface and the sound interpretation of the sugested notes of the neural models is made using the ToneJS (https://tonejs.github.io/) library (Figure 2). For music composition, the user must respect the letter notation, in which each note is written starting with its letter symbol (A to G) and octave (as number), and ending with its duration in parentheses (for example, A1(1.0)). The duration from the parentheses is represented as a multiple of the quarter

note, which means that the duration of 4.0 represents a full note. In the future, an interface will be implemented in order that the user can use it to enter notes in a piano-like way.

Figure 1. The web interface of the text generation tab

The web server logic (with its code being on the client after he accessed the web site) communicates with the app server by AJAX requests, which increases the interactivity of the application, and the content of the requests is represented in the JSON format which is a characteristic of REST applications.

The architecture of the application (presented in Figure 3) consists of a nodeJS (https://nodejs.org/en/) web server and a waitress REST application server. By using the REST (https://restfulapi.net/) architecture, the application server is less stressed by the user requests and this is an important feature as it has to handle the model sugestions. On theclient side has been used the ReactJS (https://reactjs.org/) framework that handles the logic and CoreUI (https://coreui.io/react/) for ReactJS as the theme for the site, so that future improvements will be added easily. Separate modifications of the theme had been made using HTML, CSS and Reactstrap (https://reactstrap.github.io/).

The architecture of the application (presented in Figure 3) consists of a nodeJS (https://nodejs.org/en/) web server and a waitress REST application server. By using the REST (https://restfulapi.net/) architecture, the

application server is less stressed by the user requests and this is an important feature as it has to handle the model sugestions. On theclient side has been used the ReactJS (https://reactjs.org/) framework that handles the logic and CoreUI (https://coreui.io/react/) for ReactJS as the theme for the site, so that future improvements will be added easily. Separate modifications of the theme had been made using HTML, CSS and Reactstrap (https://reactstrap.github.io/).

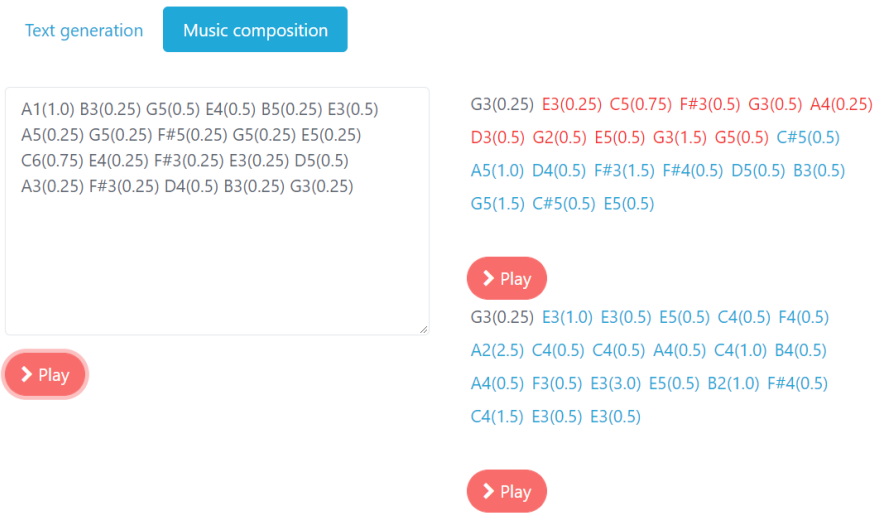


Figure 2. The web interface of the music composition tab

The app server logic is implemented using python and Flask (https://flask.palletsprojects.com/en/1.1.x/) web framework. The content of the request from theclient side is directed to the TensorFlow (https://www.tensorflow.org/) sugestion models and the result is handled back to the client. In the case of music composition, before the models are trained, for the data ingestion and data preparation of the corpus that is represented as MIDI files, has been used the Music21 library http://web.mit.edu/music21/.

The language models used for text generation are represented by two neural networks that have been trained using TensorFlow. The first one is a bidirectional GRU with 1000 internal units (that represent the dimensionality of the output of the GRU neural network) and a dense layer of interconnected

neurons with a softmax activation at the end of the network. At the input of the network, an embedding layer has been used that takes as input a matrix of hot-encoded words (one batch of 50 words for each iteration) and outputs an encoded representation of the input words that has the size of 250. The large batch used for each iteration is increasing the efficacy of the model and it also eliminates the possibility that the suggested text will have repetitive strings, because 50 identical consecutive words is almost impossible to be found twice within the same text. The model has been trained using Adam optimization with a learning rate of 0.0001 that minimizes a categorical cross-entropy loss.

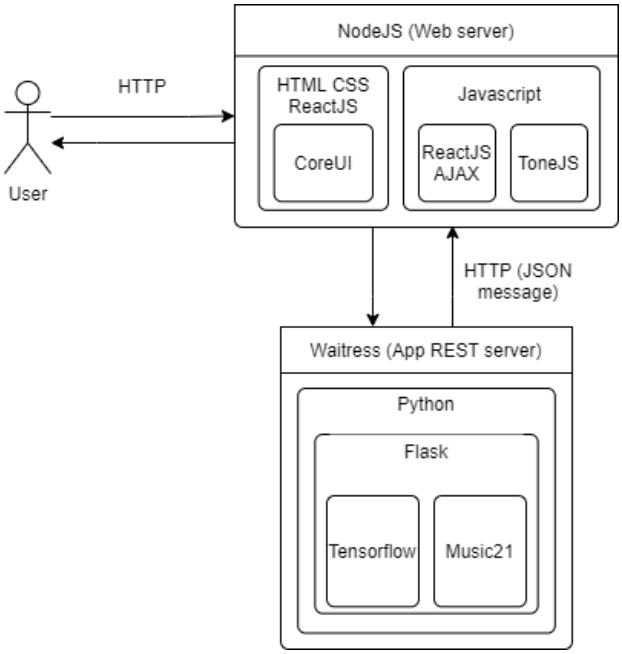


Figure 3. The architecture of the application

The second model is similar to the first one, but instead of a bidirectional Gated Recurrent Units (GRU), two successive GRU layers with 1000 internal units each have been used. The corpus that was used for training the models is the tiny Shakespeare dataset that has 40000 lines of text extracted from

different Shakespeare plays[2].

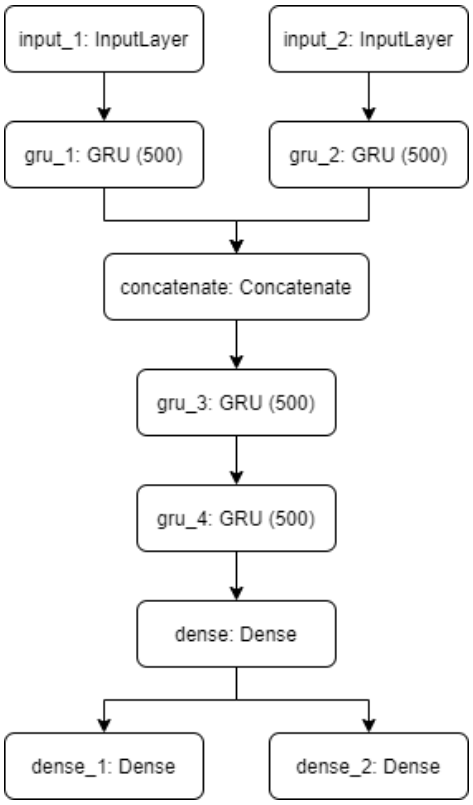


Figure 4. The first neural network model used for music composition

The first model (represented in figure 4) takes as input two batches of length 50. The first batch represents the first set of 50 notes and the second one represents their corresponding durations. Each batch is entered into an GRU layer with 500 internal units and after this the outputs of all states from each GRU layer are concatenated by their last dimension which means that each encoded note is concatenated with its corresponding encoded duration. Then the output is entered in two consecutive GRU layers, each having 500 internal units, and in one final dense layer. At the end, the output is split into

[2] Extracted from https://raw.githubusercontent.com/karpathy/char-rnn/master/data/tinyshakespeare/input.txt at 6.07.2020

two dense layers with softmax activation, where the output of the first layer is corresponding to the sugested output note for the input sequence of notes and the output of the second layer is corresponding to the sugested note duration.

The second model is the same as the first one with the exception that instead of two consecutive GRU layers has been used one bidirectional GRU layer with 500 internal units. Both neural models have been trained on a corpus that contained the following Johann Sebastian Bach compositions in MIDI format[3]: English suites, Goldberg variations, sinfonias, and inventions.

4. Experiments

The text that the user inserts into the input area of the interface of the application is passed to the language models from the server, and the resulted sugested text is then passed back to the web interface where it is displayed to the user.

On the web interface, the user can select how much text from the sugested text she wants to append to her written words. An example of a user that writes text by interaction with the language models is the following one:

- The user types: “We”.
- The user selects from the output of the first language model the first three words: “think the world”. The text in the text area now becomes: “We think the world”.
- The user selects from the output of the second language model the first four words: “and norfolk and thou”. The text in the text area now becomes: “We think the world and norfolk and thou”.
- The user selects from the output of the first language model the first eight words: “shalt to the prince my lord I know”. The text in the text area now becomes: “We think the world and norfolk and thou shalt to the prince my lord I know“.
- The user modifies the text from the text area to: “We think the world and norfolk and thou shalt to the prince my lord we know”.
- The user selects from the output of the first language model the first

[3] Extracted from https://www.mutopiaproject.org/cgibin/make-table.cgi?collection=bachis at 6.07.2020

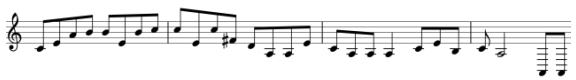
two words: “his remedy”. The text in the text area now becomes: “We think the world and norfolk and thou shalt to the prince my lord we know his remedy”.

For music composition, the user must respect the letter notation, presented in the previous section. The following experiment shows one example of composed music that uses human neural networks interaction. The first input from the user represents the first notes of the melody "Stairway to Heaven" by Led Zeppelin, which shows that the models can still sugest well enough when they face music from an entirely different genre (we may say that the application generates music in the style of Johann Sebastian Bach starting from a rock theme):

The user types:

```
“C4(0.5) E4(0.5) A4(0.5) B4(0.5) B4(0.5) E4(0.5)
B4(0.5) C5(0.5) C5(0.5) E4(0.5) C5(0.5) F#4(0.5)
D4(0.5) A3(0.5) A3(0.5) E4(0.5) C4(0.5) A3(0.5) A3(0.5)
A3(1.0) C4(0.5) E4(0.5) B3(0.5) C4(0.5) A3(2.0) A2(0.5)
A2(0.5)”,
```

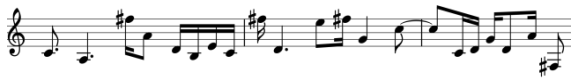
which is a textual codification of the following stave:



The user selects from the output of the first language model the notes:

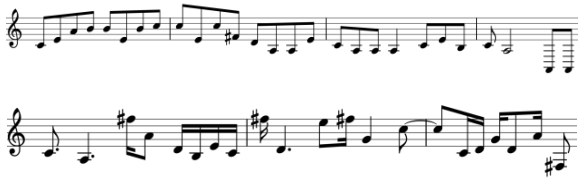
```
“C4(0.75) A3(1.5) F#5(0.25) A4(0.5) D4(0.25) B3(0.25)
E4(0.25) C4(0.25) F#5(0.25) D4(1.5) E5(0.5) F#5(0.25)
G4(1.0) C5(1.0) C4(0.25) D4(0.25) G4(0.25) D4(0.5)
A4(0.25) F#3(0.5)”,
```

that means:



The notes in the text area now become:

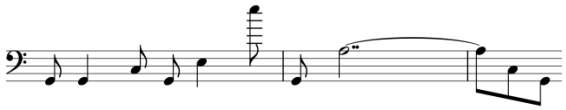
```
“C4(0.5) E4(0.5) A4(0.5) B4(0.5) B4(0.5) E4(0.5)
B4(0.5) C5(0.5) C5(0.5) E4(0.5) C5(0.5) F#4(0.5)
```



D4(0.5) A3(0.5) A3(0.5) E4(0.5) C4(0.5) A3(0.5) A3(0.5)
A3(1.0) C4(0.5) E4(0.5) B3(0.5) C4(0.5) A3(1.0) A2(0.5)
A2(0.5) C4(0.75) A3(1.5) F#5(0.25) A4(0.5) D4(0.25)
B3(0.25) E4(0.25) C4(0.25) F#5(0.25) D4(1.5) E5(0.5)
F#5(0.25) G4(1.0) C5(1.0) C4(0.25) D4(0.25) G4(0.25)
D4(0.5) A4(0.25) F#3(0.5)".

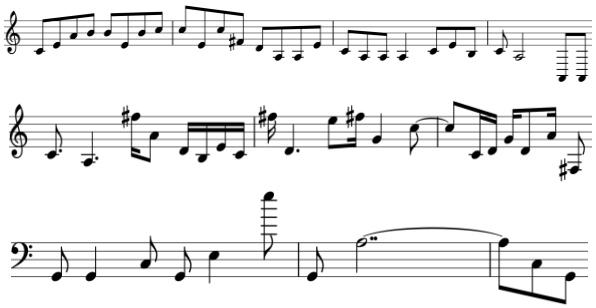
The user selects from the output of the second language model the notes:

"B2(0.5) B2(1.0) C3(0.5) B2(0.5) E3(1.0) E5(0.5)
B2(0.5) A3(2.0) C3(0.5) B2(0.5)".



The notes in the text area now become:

"C4(0.5) E4(0.5) A4(0.5) B4(0.5) B4(0.5) E4(0.5)
B4(0.5) C5(0.5) C5(0.5) E4(0.5) C5(0.5) F#4(0.5)
D4(0.5) A3(0.5) A3(0.5) E4(0.5) C4(0.5) A3(0.5) A3(0.5)
A3(1.0) C4(0.5) E4(0.5) B3(0.5) C4(0.5) A3(1.0) A2(0.5)
A2(0.5) C4(0.75) A3(1.5) F#5(0.25) A4(0.5) D4(0.25)
B3(0.25) E4(0.25) C4(0.25) F#5(0.25) D4(1.5) E5(0.5)
F#5(0.25) G4(1.0) C5(1.0) C4(0.25) D4(0.25) G4(0.25)
D4(0.5) A4(0.25) F#3(0.5) B2(0.5) B2(1.0) C3(0.5)
B2(0.5) E3(1.0) E5(0.5) B2(0.5) A3(2.0) C3(0.5)
B2(0.5)".



5. Conclusions

The examples presented in the paper show that artificial intelligence has the advantage of speed in the generation process, which humans don't have, but in most cases, it lacks humans' ability to create text with significance.

However, the examples show that a hybrid approach with humans interacting with intelligent algorithms can produce acceptable results by using the strengths of both. Another remark is that music generation with artificial intelligence provides better results than in the case of text (however, if the listener is open to experimental musical creations). This fact may be explained by the more fixed structure of text and that music needs less coherence than texts. However, if a listener of the musical piece expects a simple, pleasant melody, probably s/he will not be satisfied.

The next challenge is to implement more powerful content generation algorithms and to find a metric for the quality of the final human and neural network generated content, which will establish if the further improvements of the application are going in the right direction. This type of metric should follow artistic benchmarks such as symmetry or other esthetic criteria.

Other improvements can be made on the data that is used for training the algorithms and the type of models that are used for learning this data. For music composition, the language models should also be able to compose using rhythm, stresses, rests, musical timbre, velocity, beats and multiple voices (polyphony, which should be considered both for music and text (Trausan-Matu, 2020). For text generation, punctuation and paragraphs should be taken into consideration, as a priority.

References

Csikszentmihalyi, M. (1996). Creativity, Flow and the Psychology of Discovery and Invention. Harper Perennial.

de Bono, E. (2009). Lateral Thinking: A Textbook of Creativity. Penguin Group.

Bruce, J.L. (1995). Composing with genetic algorithms. In *Proceedings of the International Computer Music Conference*, Banff Alberta.

Engel Jesse, Resnick Cinjon, Roberts Adam, Dieleman Sander, Eck Douglas, Simonyan Karen, Norouzi Mohammad. (2017). Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. Retreived from https://arxiv.org/abs/1704.01279 at October 5, 2020.

Oprișan, A., Trausan-Matu, S. (2013). Creativity Stimulation Tool. *Annals of the Academy of Romanian Scientists Series on Science and Technology of Information*, Volume 6, Number 1, 2013, pp. 63-83

Roberts Adam, Engel Jesse, Raffel Colin, Hawthorne Curtis, Eck Douglas. (2019). *A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music.* Retreived from https://arxiv.org/abs/1803.05428 at October 5, 2020.

Trausan-Matu, S. (2020) The Polyphonic Model of Collaborative Learning, in Mercer, N., Wegerif, R., & Major, L. (eds.), *The Routledge international handbook of research on*

dialogic education, New York, NY : Routledge, pp. 454-468, https://doi.org/10.4324/9780429441677

Magenta (2020) https://magenta.tensorflow.org/2016/12/16/nips-demo, last accessed on October 5, 2020.

Diederik P. Kingma and Jimmy Lei Ba. (2014) Adam: A method for stochastic optimization. *arXiv*:1412.6980v9.