Musical Style Learning: Generating new Beatles' songs

Paolo Fantozzi, Luigi Laura
Sapienza, Uninettuno
paolo.fantozzi@diag.uniroma1.it,luigi.laura@uninettunouniversity.net

Abstract

Recently we are witnessing an increasing amount of *artistic* application of Deep Learning techniques: so far, the research focused mainly in image creation (see, e.g., the very popular Image Style Transfer [Gatys *et al.*, 2016]), probably due to the success of Convolutional Neural Networks for image manipulation, but there are growing areas of research devoted to other applications.

In this paper, we deal with the generation of new songs, based on the style of a specific artist (in our case, a band (we probably should say 'the' band): the Beatles. The generation of a new song requires the generation of several parts that, needless to say, should play along well in order to have a nice song: the first elements are, obviously, music and lyrics and later there are the musical arrangement, the musical instruments and, last but not least, the singing.

In the following we discuss the first step of our long term project, aimed at the generation of new Beatles' song: in particular, this step is focused on lyrics generation, and it is based on a Transformer architecture, using a pre-trained model of 345 millions parameters. Our results are promising: despite the relatively small size of the network used, we had some songs that can be considered nice, although others were definitely ugly. Our code is freely available on Github.

1 Introduction

The very last years have seen the rise of ubiquitous Deep Learning approaches for solving a variety of tasks, including *artistic* ones: we mention the popular Image Style Transfer By Garys, Ecker and Bethge [Gatys *et al.*, 2016], that showed how to use feature representations from Convolutional Neural Networks (CNNs) to transfer image style between arbitrary images. Lecoutre et al. [Lecoutre *et al.*, 2017] used CNNs to recognize the art style of paintings. These results should not be surprising, since in some sense the artistic style of a painter is something that even a pigeon can recognize, as in the famous experiment by Watanabe et al. [Watanabe *et al.*, 1995], where pigeons were able to distinguish paintings by Monet and Picasso.

Here we deal with the problem of the generation of new songs, based on the style of a specific artist that, in our case, is probably the world's most iconic music band: the Beatles. The generation of a new song requires the generation of several parts that, needless to say, should play along well in order to have a nice song: the first elements are, obviously, music (both harmony and melody) and lyrics and later there are the musical arrangement, the musical instruments and, last but not least, the singing.

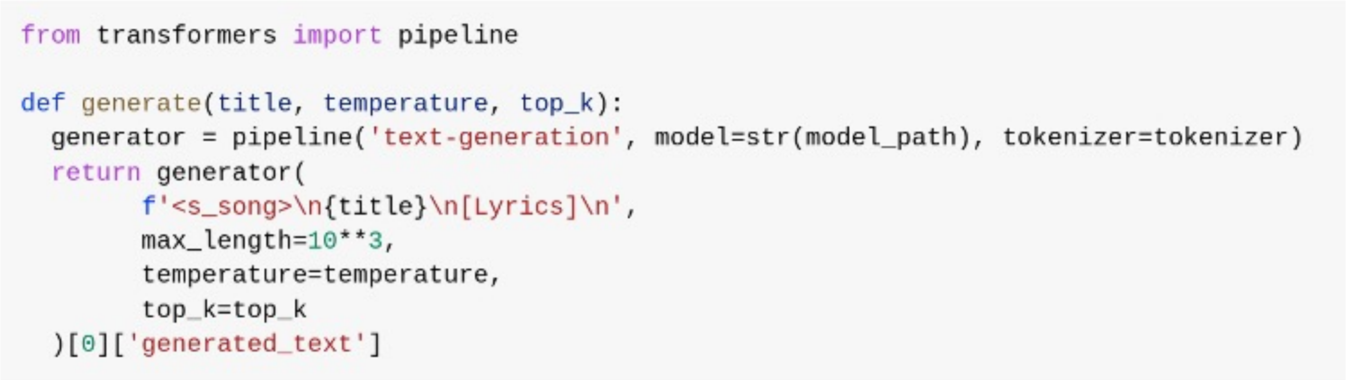
In this paper we discuss the first step of our project aimed, as already mentioned, at the generation of new Beatles' song: in particular, this step is focused on lyrics generation, and it is based on a Transformer architecture [Vaswani *et al.*,], using a pre-trained model of 345 millions parameters. Our results are promising and our code is available on Github.

2 Related work

There are several approaches for music generation using Deep Learning techniques, we refer the interested reader to the surveys of Briot and Pachet [Briot e Pachet, 2020] and Briot et al. [Briot *et al.*, 2017].

Since we focus on lyrics, the aim of this research is to generate text that follows some kind of guide lines (i.e., it should be similar to lyrics) having just a few of samples. The neural networks for sequence to sequence generation are a field very active in the recent years.

Even if our approach seems to be promising it can't compete with the computing brute force of the recent approaches. Indeed, since the introduction of the Transformer architecture in [Vaswani *et al.*, 2017], the size of the natural language models seems to increase at least twice a year. From the first implementation of BERT in [Devlin *et al.*, 2019] and GPT in [Radford *et al.*, 2018], each actor in the field tried to expand the capabilities of the models increasing their size more and more. The first model to start the increasing trend was, indeed, GPT [Radford *et al.*, 2018] with 110M parameters, then Google released BERT [Devlin *et al.*, 2019] followed, 4 months later, by Open-AI GPT-2, presented in [Radford *et al.*, 2019] with 1.5B of parameters. Six months later NVIDIA joined to this field of research presenting MegatronML in [Shoeybi *et al.*, 2019] with 8.3B parameters, and after more six months also Microsoft presented a transformer-based model in [Inc., 2020 accessed October 1 2020] with



```python
from transformers import pipeline

def generate(title, temperature, top_k):
  generator = pipeline('text-generation', model=str(model_path), tokenizer=tokenizer)
  return generator(
        f'<s_song>\n{title}\n[Lyrics]\n',
        max_length=10**3,
        temperature=temperature,
        top_k=top_k
  )[0]['generated_text']
```

Figure 1: Text generator definition

17B parameters. Actually the largest model is M6, developed by Alibaba DAMO Academy (i.e., the R&D branch of Alibaba): a large multimodal, multitasking language model with 10 trillion parameters.

3 Our approach

The goal of the project is to generate new Beatles' songs, using as training set the songs actually recorded by Beatles. In this paper, as we mentioned, we focus on the lyrics. We want to use recent techniques of sequence to sequence neural networks that has reached state of the art results but, at the same, time, we would like to test the effectiveness of relatively small models. Indeed, the recent models published by OpenAI, called GPT-2, seem to be aligned to our goal: even if the full trained model has not been released, we opted to use a smaller model called 345M, which is the dimension of the parameters in the neural network. This model should have results not as good as the full-sized model, but it should be enough to create a proof of concept with some interesting results.

3.1 Data collection & source(s)

The train dataset used in this project consists in all the songs recorded by Beatles. We used the Genius API (https://docs.genius.com/) to retrieve all the lyrics. The entries are all stored in local to be analysed and preprocessed.

All the lyrics are inserted in a single collection. So they are compared, one by one, by title to remove clear duplicates. Then the lyrics are analyzed by some metrics (for instance the length) to remove instrumental songs. The lyrics obtained after the preprocessing are 380.

Since that the songs recorded by Beatles are just 150 (including instrumentals), then it means that we have many duplicates or text that is not a lyric. Taking some samples, it seems that there are some entries that are not songs, but facts about songs and Beatles in general. Since we are using a pre-trained neural network we decided, for this test, to leave this texts in the dataset, to be used for the training as well. The noise given by these data should impact only on the layout of the response, as we will see in the results.

3.2 Models & Methods

As mentioned before, we used the GPT-2[Radford *et al.*,] neural network, based on the Transformer architecture[Vaswani *et al.*,]. Using the technique of transfer learning we used a pre-trained model of 345 millions parameters, downloaded from https://huggingface.co/. Then we refined the model using the dataset we created.

The resulting model should create samples based both on the content and the layout of the train dataset, even if still using the knowledge represented in the 345M model.

3.3 Software/Hardware Toolkit

We used the virtual environment provided by Google on their servers, called Google Colaboratory (also known as Colab). This environment provides jupyter notebooks with GPU support and Tensorflow already installed. We also used a fork of the official implementation of GPT-2 provided at https://github.com/Tenoke/gpt-2.git which includes the feature of refine training of a model.

4 Results

In Figure 1 it is shown the code of the main method for the text generator definition; in particular we can see that it takes three parameters: the first one is the maximum length of the lyric, the other two are temperature and top_k. The first one, temperature, it is the temperature of Boltzmann distribution to scale the probabilities of the next words; if it is close to 0, the model will become deterministic, repeating always the same words; higher temperature results in more random completions.

The other parameter, `top_k`, chooses between the most probable K next words; if it is equal to zero it means all the words can be considered. If we choose a low value of `top_k` we might loop in the same small set of words, whilst a high value lead to words that are out of topic, as seen in Figure 3.

All our code is available at https://github.com/paolofantozzi, it runs in the Google Colab environment so feel free to use it to generate new lyrics from your favorite artist (you only need to change the query to the lyrics database Genius).

In Figure 2 it is possible to see an example of a lyric generated by the approach described. Since we are in an *artistic*

```
Love of my life
[Lyrics]
[Verse 1]
Nothing can come between us
When it gets dark I tow your heart away
Never thought of act of love I’ve ever known

[Chorus]
For if my love had lasted a day
We might have broken down
If we’d seen each other before

[Verse 2]
I can’t forget the time or place
Where we just met, she’s just the girl for me
And I want all the world to see we’ve met
Mmm-mmm-mmm-da-da

[Chorus]
Love of my life

[Verse 3]
Nothing can come between us
When it gets dark I tow your heart away
Never thought of act of love I’ve ever known

[Chorus]
For if my love had lasted a day
We might have broken down
If we’d seen each other before

[Verse 4]
I can’t forget the time or place
Where we just met, she’s just the girl for me
And I want all the world to see we’ve met
Mmm-mmm-da-da-da

[Chorus]
Love of my life

[Verse 5]
I can’t forget the time or place
Where we just met, she’s just the girl for me
And I want all the world to see we’ve met
Mmm-mmm-da-da-da

[Chorus]
For if my love had lasted a day
She would never have said a word
To me, she’s just the girl for me
Just the thing that makes me tick
I think of her almost all the time

[Verse 6]
It’s just a feeling in my heart
Like an ocean of tears waiting to be lit
Till I find the one I really love
And I want all the world to see we’ve met
Just the thing that makes me tick
Oh-oh-oh-oh
```

Figure 2: An example of a lyric generated

field, it is indeed difficult to measure the quality of the obtained result. The song shown in Figure 2 is one of the many that, to our personal judgment, looked like a reasonable lyric, even if others might disagree. We had many results that are definitely ugly (we do not report them for the lack of space and because they are ugly).

5 Conclusion

In this paper we briefly presented an approach to generate new lyrics learning the *artistic style* of an artist or a band. The approach is based on sequence to sequence neural networks, that is a research area in which, in the last years, there has been a race for the largest model, a race in which nowadays only few players can compete, due to the scale of resources needed.

Our question, and our experiment, is whether can we obtain interesting results using small scale pre-trained neural networks, such as the ones freely available from Hugging Face (https://huggingface.co/). Our code is freely available and can be used to generate lyrics of *your* favourite artist, simply by changing the query to the lyrics database Genius (https://genius.com/).

Ideally, this simple tool can be used also by lyrics writers, that after training it with their own lyrics, can have an assistant to help them in the writing process: thus, it can be an example of *augmented creativity* application. Indeed, as we mentioned, we had both nice and ugly results; a professional lyric writer can generate many of them and then mix, adapt, rewrite and change up to his own requirements.

Our next step will be devoted to the development of music (both harmony and melody). The road to a new Beatles' song is a long one.

References

[Briot *et al.*, 2017] Jean-Pierre Briot, Gaëtan Hadjeres, e François-David Pachet. Deep learning techniques for music generation–a survey. *arXiv preprint arXiv:1709.01620*, 2017.

[Briot e Pachet, 2020] Jean-Pierre Briot e François Pachet. Deep learning for music generation: challenges and directions. *Neural Computing and Applications*, 32(4):981–993, 2020.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, e Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, e Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.

[Gatys *et al.*, 2016] Leon A Gatys, Alexander S Ecker, e Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

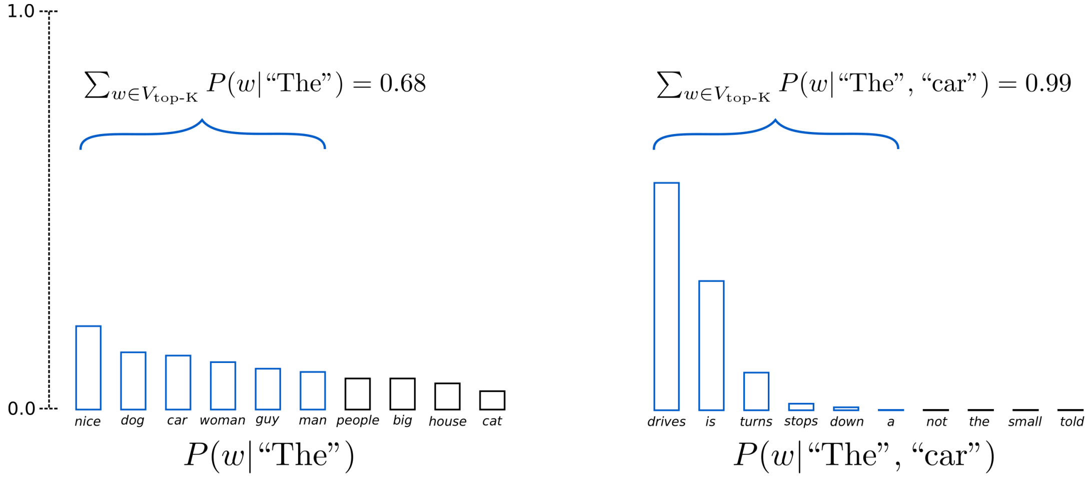


Figure 3: Top k sampling effect (from https://huggingface.co/blog/how-to-generate)

[Inc., 2020 accessed October 1 2020] Microsoft Inc. *Turing-NLG: A 17-billion-parameter language model by Microsoft*, 2020 (accessed October 1, 2020). https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

[Lecoutre *et al.*, 2017] Adrian Lecoutre, Benjamin Negrevergne, e Florian Yger. Recognizing art style automatically in painting with deep learning. In *Asian conference on machine learning*, pages 327–342. PMLR, 2017.

[Radford *et al.*,] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, e Ilya Sutskever. Language Models are Unsupervised Multitask Learners. Technical report.

[Radford *et al.*, 2018] Alec Radford, Karthik Narasimhan, Tim Salimans, e Ilya Sutskever. Improving language understanding by generative pre-training. *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf*, 2018.

[Radford *et al.*, 2019] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, e Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8), 2019.

[Shoeybi *et al.*, 2019] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, e Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using gpu model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[Vaswani *et al.*,] Ashish Vaswani, Google Brain, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, e Illia Polosukhin. Attention Is All You Need. Technical report.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, e Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, e Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008, 2017.

[Watanabe *et al.*, 1995] S Watanabe, J Sakamoto, e M Wakita. Pigeons' discrimination of paintings by monet and picasso. *J. Exp. Anal. Behav.*, 63(2):165–174, March 1995.