# Natural Language Processing - Disco Polo generator

Mateusz Marzec, Paulina Pacyna, Mateusz Szysz

29 May 2022

## Contents

# 1 Introduction

The goal of the project is to build a model capable of generating lyrics for disco polo songs. We were inspired by various examples of such work[1][2][3] ([UVM21]). We plan to collect the data from Tekstowo.pl by scraping most popular Disco Polo bands. Then we want to use pretrained language model by finetuning it on our dataset. In this way we want to learn the model to generate Disco Polo. It this won't be enough we plan to enhance the process of generating songs. We expect many songs to be bad, so we want to create a model capable of scoring how good the generated song is, and based on this score present only songs with highest obtained score.

# 2 Problem definition

Our goal is to build a model capable of generating Disco Polo songs. We expect some sort of input (i.e. initial verse) to start the generating process. We have approach the problem using language models - the model that given some context produces the probability distribution over all possible words (tokens). After finetuning on specific task, language models are capable of creating various outputs like answers to given questions or poetry.

# 3 Methods and approach

For the task of generating lyrics fro disco polo songs, we found the Generative Pre-Trained model (GPT-2) the most suitable. GPT-2 is a neural network, pretrained on a vast amount of data, but not designed to a specific task. Some of the most important features of this architecture are:

- attention mechanism - the network learns to focus on a part of input sentence when generating a certain part of the output sentence. Pair of indexes $k_{ij}$, which means attention paid to *ith* input word when generating *jth* output word are produced by a small network, which learns them from the training data.
- decoder architecture - build up from transformer decoder blocks.

We trained and tested our models on Google Cloud Platform.

[1] Medium article

[2] Towards Data Science article

[3] Polski poeta AI

### 3.1 Finetuning

We only finetune our models (no learning from zero). We use early stopping to prevent the models from overfitting. As a metric we use perplexity calculated on validation set. When the perplexity stops decreasing (the perplexity from current epoch is higher than average perplexity from 2 previous epochs) we stops the training process.

## 4 Results

### 4.1 Data collection

The data was scraped from popular online website Tekstowo.pl. This website contains a myriad of songs and their lyrics. In our problem we were mainly interested in Disco Polo songs. The initial step was to collect the names of popular Disco Polo bands. Next, having the names were we able to scrape all the songs associated with a given band name. This resulted in obtaining 6798 songs from 134 different bands.

### 4.2 Data preprocessing

The data had to be preprocessed. As the first model was trained on verses and the second on songs the preprocessing steps required for both models were slightly different. For both datasets we were interested in adding tokens to improve the generation part. Introduced tokens with their meanings are presented below:

- <EOST> - end of stanza
- <RBEG> - beginning of the chorus
- <REND> - end of the chorus
- < |startoftext| > - beginning of the song
- < |endoftext| > - end of the song

Each of the tokens above was used to augment the song generating process. After preprocessing the data was splitted into train, validation and test set (in following proportions: 80%, 10%, 10%).

## 4.3 Finetuning existing models

We tried to finetune 2 models. They are in fact the same, but were pretrained on different text data. The first one - GPT-2 model was taken from gpt_simple[4] library. It was finetuned on both verse and songs datasets (we have obtained 2 models as a result). The second - papuGaPT2 from transformers[5] library.

### 4.3.1 papuGaPT2

Due to computational issues we were not able to finetune papuGaPT2 model. One epoch would took around 50 hours on our machine. Due to this fact we did not report any results for papuGaPT2 model.

### 4.3.2 GPT-2 verse

One of the initial idea was to train the model on verses with tokens at the end of each line instead of songs. We tried to fit the model with different numbers of steps. However, each time the model worked very badly and consecutive verses were not linked at all.

### 4.3.3 GPT-2 song

Model was finetuned on songs training dataset. Dataset consists of nearly 5000 Disco Polo songs. After each 200 steps (one step is one forward and backward pass thorough the network, performed with batch of inputs) we calculate the perplexity on validation set. Unfortunately the library did not provide such functionality. Some solution was adapted but it worked poorly. We had to limit the amount of text on which the perplexity was calculated to smaller sample, due to computational issues (calculating perplexity on whole validation test took to long). For unknown reasons the code stopped working after a while. We were not able to solve this issue. Additionally the stopping rule stopped the model learning quite quickly. The results are presented on Figure 1.

[4] gtp simple python library

[5] papuGaPT2 on transformers library

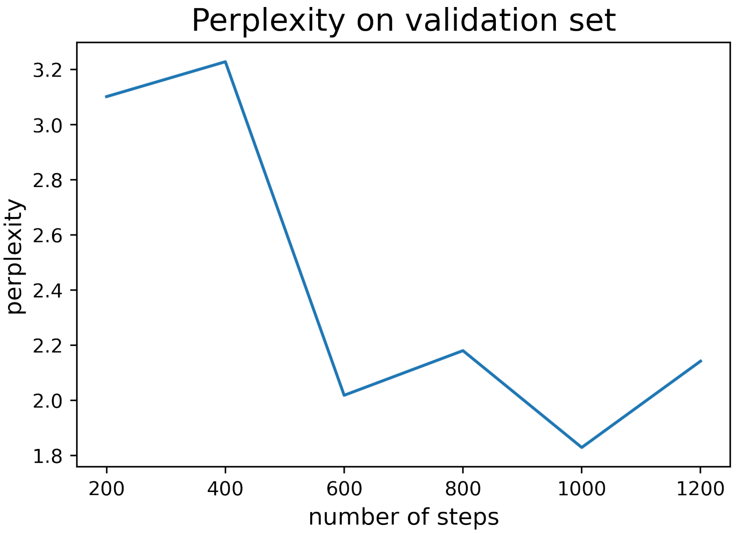

Figure 1: Perplexity calculated on subset of validation set during training with stopping rule.

We suspect that the stopping rule overreacted and the model should be trained for a bit longer period of time. Consequently the model was then trained for 9000 steps, at each step calculating perplexity. The results are shown on Figure

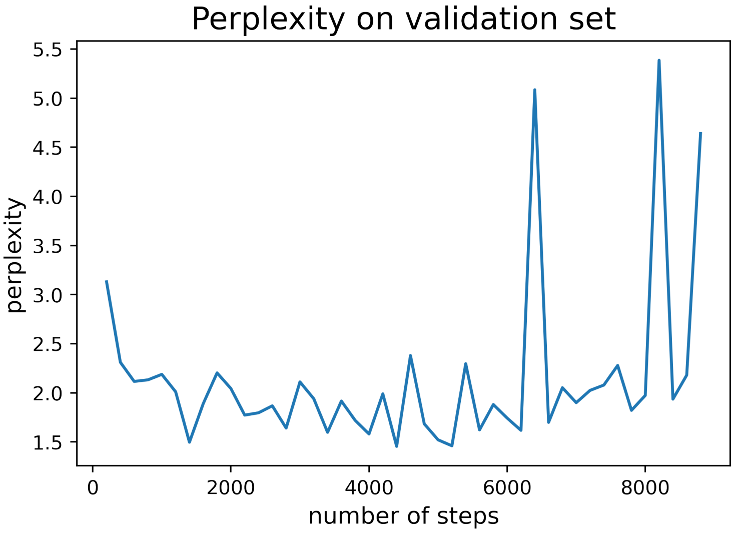

Figure 2: Perplexity calculated on subset of validation set during training without stopping rule.

The perplexity seems to drop fast and then fluctuate for some time. After around 6000 steps we see random spikes. The number of 1200 steps seems to be too low, so out guess is that perplexity (calculated on subset of validation set) is not reliable metric in our case.

## 4.4 Augmenting the generation process

To force the model to generate outputs with song alike structure we have embedded some rules into generating function available in library we have used. We tried to use tokens described in section 4.2 in order to make model generate more structured outputs. Model starts with short context (one verse) and generates the output until it produces token representing start of chorus (so the first part of generated text should be first stanza). Then the context is expanded with newly generated text and token representing start of chorus. Another rule is applied - model stops generation process when it outputs the token representing end of chorus. This way we should have a piece of text that looks like stanza and chorus. The next steps are followed until desired length of text is achieved. At each step we expand the context with generated text and generate the some portion of text until the end of stanza token will appear. At each step we check simple rule - we expect the length of generated chunk to be above some threshold (like 40 characters).

## 4.5 Distinguishing real and generated disco-polo

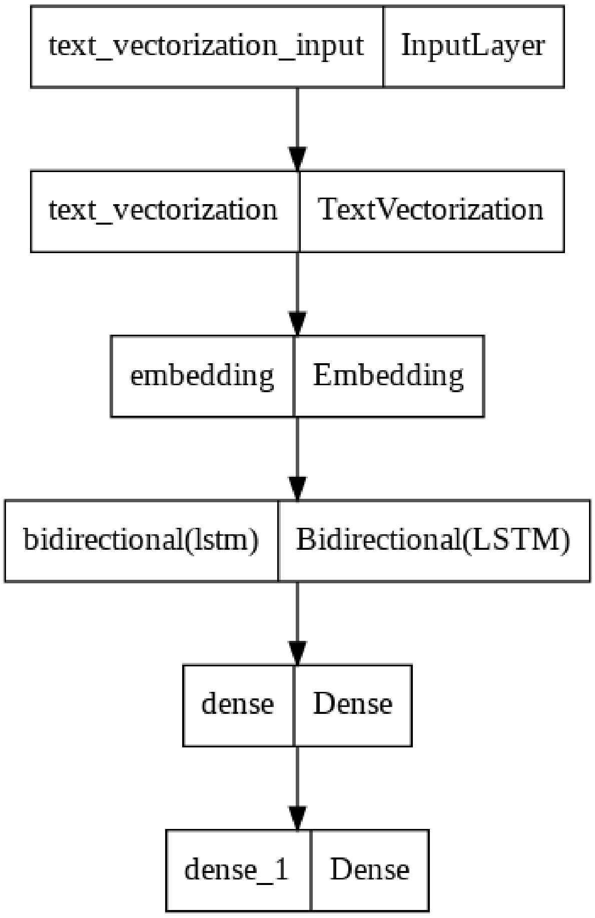

Figure 3: Scheme of the model distinguishing the real and the generated disco-polo.

In order to choose the best subset of generated songs we created the model that distinguishes real disco-polo from the generated one. The scheme of of the architecture is shown in fig. 3. Even though the architecture is very simple, it allowed to get the 99% accuracy on the independent test set. Creating this model was aimed at filtering all the generated songs and finding ones that resemble the real ones the most. Examples of generated songs are shown in the next subsection.

## 4.6 Similarity to disco-polo classifier

To obtain a score of how a song fits into disco polo genre we trained a classification model. We scraped disco-polo and other genres from Tekstowo.pl. The dataset consisted of 2 columns: song and label(1 if the song was discopolo, 0 otherwise). The classes in the dataset were balanced and the songs used to train the model were tokenized and preprocessed using the same methods as in our main model. We used a `bert-base-cased` model from hugging-face. We tried to use polish versions of pretrained models, but those models were enforcing a batch size of 1024, which caused a out of memory error. We evaluated our model on the test dataset and we obtained 84% accuracy.

## 4.7 Repeatability score

As we struggled with some of the verses being repeated, we created a heuristic score of repeatability of a song. We calculated how many times every N-gram was repeated, and the score was equal to minus average of number of occurences of every N-gram. We used 4-grams to score the songs.

## 4.8 Sample of created songs

In this subsection we showed a few examples of songs created from ”Przez twe oczy” context.

Przez twe oczy z toba wziałem
Spojrzeje za rok, spojrzeje za reke
Jak nie kochałaś tak jak Ty
Gdy noc jeszcze raz, gdy noc jeszcze raz

Tak miało być, tak miało być
Wolnym być, wolnym być
I znów rozpalił sie
Wolnym być, wolnym być
I znów rozpalił sie
Wolnym być, wolnym być
I znów rozpalił sie

Chciałem być i jak przytulić
I od dawna to nie żałuje
Odeszłaś, zapłaciłem
I znowu wyjawiłaś sie
I znowu wyjawiłaś sie
I znowu wyjawiłaś sie

Tak miało być, tak miało być
Wolnym być, wolnym być
I znów rozpalił sie
Wolnym być, wolnym być
I znów rozpalił sie
Wolnym być, wolnym być
I znów rozpalił sie

Tak miało być, tak miało być
Wolnym być, wolnym być
I znów rozpalił sie
Wolnym być, wolnym być
I znów rozpalił sie

Przez twe oczy, oczy czarne myśli
I tak za nim zostana jest
Bo na zawsze już wybrałem, miałem
Nie liczł na nic, liczył na noc

Ref.:
A teraz chodź kochanie, weź wegryź w miejscu
Na nic nasz wszystkie dni, gdzie najszybciej
Niech słysze szum, ciepłych słów
Jak kiedyś poznam szcześcia mam

Nie wiem co sie dzieje w mej głowie
Wypijemy gdzieś, otwieraja sie
Na twarzy uśmiech masz
Wciaż wpadaja nam, znów pytaja mnie

Ref.: A teraz chodź kochanie, weź wegryź w miejscu
Na nic nasz wszystkie dni, gdzie najszybciej
Niech słysze szum, ciepłych słów
Jak kiedyś poznam szcześcia mam

A teraz chodź kochanie, weź wegryź w miejscu
Na nic nasz wszystkie dni, gdzie najszybciej
Niech słysze szum, ciepłych słów
Jak kiedyś poznam szcześcia mam

A teraz chodź kochanie, weź wegryź w miejscu
Na nic nasz wszystkie dni, gdzie najszybciej
Niech słysze szum, ciepłych słów
Jak kiedyś poznam szcześcia mam
A teraz chodź kochanie, weź wegryź w miejscu
Na nic nasz

Przez twe oczy, oczy czarne myśli
I tak za nim zostana jest
Bo na zawsze już wybrałem, miałem
Nie liczł na nic, liczył na noc

Ref.:
Bo właśnie ty, tylko ty
Odpłyniesz w swój uśmiech
Ciepłych słów
Wkrótce przeminie lato, jeszcze raz
Niech sie spełni żyje, w niej jest źle

Już zabawa i ten film
I tak zapodaje, że
Niech sie spełnia nasza miłość
I tak za nim zostana jest
Bo na zawsze już wybrałem, miałem
Nie liczł na nic, liczył na noc

Ref.:
Bo właśnie ty, tylko ty
Odpłyniesz w swój uśmiech
Ciepłych słów
Wkrótce przeminie lato, jeszcze raz
Niech sie spełni żyje, w niej jest źle

Bo właśnie ty, tylko ty
Odpłyniesz w swój uśmiech
Ciepłych słów
Wkrótce przeminie lato, jeszcze raz
Niech sie spełni żyje, w niej jest źle

Ref.:
Bo właśnie ty, tylko ty
Odpłyniesz w swój uśmiech
Ciepłych słów
Wkrótce przeminie lato, jeszcze raz
Niech sie spełni żyje, w niej jest źle

Bo właśnie ty, tylko ty
Odpłyniesz w swój uśmiech
Ciepłych słów

Każda piekniejsza jak w moich reka
To jest moja od wode
Tyle uśmiech, że szalone łyć
Że meku o Twym
Zawsze Twych oczu miałem, ma ja
Tak urok kiedyś w kro
Nie chcesz na to znowu mój
Gdy odbierasz miałem rozpalone
Zawsze Twych ciało nam wciaż
Jestem ja strone
By zgasłaś mój trwać
Wszystko serce mam naszych (2 x)
Nie liczy sie i nie spotkam czuje
Bedziemy wciaż słów, ja już czekałem
Nie bede bede sie (2 x)
Niejak róży, co najlepszy
Niech to jest tam
Nie bede bede taka tak

## 5 Summary

## References

[Vas+17] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[UVM21] David Uthus, Maria Voitovich, and RJ Mical. “Augmenting poetry composition with verse by verse”. In: *arXiv preprint arXiv:2103.17205* (2021).