**Student: Xiaochi Li**

**Student number: 21211712**

**Lecturer: Dr. Tony Veale**

|  |
| --- |

**Module: Computational Creativity**

**UK Civil Servant Crap Producer**

1. Introduction

The project was inspired by the famous BBC TV show: yes, minister and yes, prime minister from the 1980s. Sir Humphrey served as Permanent Secretary in the Department of Administrative Affairs in Yes, Minister and as Cabinet Secretary in Yes, Prime Minister. He was a master of obfuscation and manipulation, often making lengthy statements to confuse and tire his listeners. The purpose of the project is to generate sentences in the form of Humphrey and read them out using the Humphrey sound.The project consists of two parts, the first part generates Humphrey-like words by retraining GPT2, the second part trains Sir Humphrey actor Sir Nigel Hawthorne's voice using Tacotron2 and converts the trained model into a .wav file using HiFi-GAN.

2.Methodology

2.1 GPT-2

GPT-2 (Generative Pre-trained Transformer 2) is a high-level language model developed by OpenAI,released in 2019. Based on the Transformer architecture, GPT-2 is trained on large amounts of Internet text data, enabling it to learn language patterns, syntax, and semantic understanding.[(Solaiman *et al.*, 2019)](https://www.zotero.org/google-docs/?OPfpvA) Our model is trained using the minimal version of GPT-2 (124M), and the training data are Sir Humphrey style sentences generated by gpt4.

Training data：

The training data is a .txt file with 118 entries. One part of the data is the original lines appearing in "Yes, Minister" and "Yes, Prime Minister", and the other part is the similar sentences generated by using gpt4.

The training data is formatted as a one-line sentence with a line break at the end and <|endoftext|>.

Example of training data：

You see, Minister, the intricacies of bureaucratic machinations are such that one cannot simply steamroll a policy into place without first performing a comprehensive survey of the potential repercussions, socio-economic factors, and political climate, not to mention achieving an almost harmonious consensus among the various bodies that, intentionally or not, stand to be affected by any radical changes, and that's not even mentioning the delicate dance of diplomatic discourse that we must conduct with our international counterparts.

<|endoftext|>

While the desire to streamline government operations is not in itself a questionable endeavor, it is, however, worth noting that any superficially benign intention, if rushed into action without due diligence, comprehensive impact assessment, and thorough feasibility studies, runs the risk of causing profound upheaval within the carefully calibrated ecosystem of interdepartmental dynamics and public service provision.

<|endoftext|>

Minister, as we consider any potential amendments to current statutes, it is paramount that we not only ponder upon the legislative implications and the possible permutations thereof but also take into account the interwoven fabric of historical precedents, socio-political equilibrium, and the potential for unanticipated ramifications that, while initially seeming negligible, could nevertheless evolve into sizeable issues that challenge the established harmony of our societal framework.

<|endoftext|>

Due to the small training data (only 49 kb), we found in previous tests that using a large learning rate (0.01) and a large number of training sessions would overfit the model, with the model's loss decreasing to 0.01 in the first few dozen iterations and fluctuating between 0.01 and 0.02. The output model will have almost the same form, although the sentences will be fluent and free of grammatical errors.Our final learning rate used a learning efficiency of 5e-4 and trained 75 times, with a loss of 0.04 for this model.

Examples of our model generates the following results:

In the realm of governance, Prime Minister, orchestrating policy reform isn't as uncomplicated as conducting a straightforward tune; it necessitates a harmonious blending of the public sentiment as our choir, the rhythm of interdepartmental dynamics, the melody of fiscal implications, and the chorus of international diplomacy, to create a symphony that leaves a lasting impact on the listeners.

<|endoftext|>

Minister, as we navigate the intricate process of policy-making, it's imperative to note that it's not a simple task of mapping a straightforward route; it involves a careful understanding of the terrain of public opinion, the routes of interdepartmental dynamics, the resources for the journey of fiscal implications, and the weather conditions of international diplomacy, to ensure a successful expedition.

<|endoftext|>

The pursuit of policy advancement, Prime Minister, isn't merely about putting the wheels in motion; it calls for a clear direction steered by public sentiment, the smooth operation of the gears of interdepartmental dynamics, the fuel of fiscal implications, and the navigation of the road map of international diplomacy, to ensure a smooth journey towards progress.

<|endoftext|>

Minister, in our quest for policy innovation, it's important to recognize that it's not just about sparking a new idea; it involves the tinder of public sentiment, the kindling of interdepartmental dynamics, the oxygen of fiscal implications, and the control of the blaze through international diplomacy, to ensure a controlled and beneficial flame, and not a destructive inferno.

<|endoftext|>

2.2 Tacotron2

Tacotron 2 is a neural network architecture for speech synthesis developed by Google. It's essentially a text-to-speech (TTS) system that converts written text into spoken words.[(Shen *et al.*, 2018)](https://www.zotero.org/google-docs/?bHNBDC)

The architecture of Tacotron 2 consists of two main components:

A sequence-to-sequence model with attention converts character text into spectrograms. The model uses an encoder-decoder architecture, where the encoder transforms the input text into a hidden feature representation, and the decoder generates spectrograms from these features.

A modified WaveNet model, acting as a vocoder, converts the spectrogram generated by the first part into an audible waveform, essentially producing human-like speech.

The training data for the model was completed by editing the video from "Yes, Minister" and "Yes, Prime Minister" and extracting the audio. The training data consisted of 182 .wav files ranging from 1 to 10 seconds in length and a text file corresponding to these wav files.The text file starts with /content/TTS-TT2/wavs/\*.wav| and is followed by a text comparison in the voice. All letters are capitalized, and no punctuation can appear in the sentences.

The format of the text files is as follows:

/content/TTS-TT2/wavs/1.wav|THE SUN SHINES BRIGHTLY IN THE SKY.

/content/TTS-TT2/wavs/2.wav|I ENJOY GOING FOR WALKS IN THE PARK.

/content/TTS-TT2/wavs/3.wav|SHE SMILED AND WAVED AT ME FROM ACROSS THE ROOM.

/content/TTS-TT2/wavs/4.wav|THE CAT CURLED UP ON THE WINDOWSILL AND FELL ASLEEP.

/content/TTS-TT2/wavs/5.wav|I LIKE TO EAT PIZZA WITH EXTRA CHEESE.

/content/TTS-TT2/wavs/6.wav|HE PLAYED HIS FAVORITE SONG ON THE GUITAR.

/content/TTS-TT2/wavs/7.wav|THE FLOWERS BLOOMED IN VIBRANT COLORS.

/content/TTS-TT2/wavs/8.wav|I WENT TO THE STORE TO BUY SOME MILK AND BREAD.

/content/TTS-TT2/wavs/9.wav|THE DOG WAGGED ITS TAIL AND LICKED MY HAND.

/content/TTS-TT2/wavs/10.wav|SHE LAUGHED AT THE FUNNY JOKE.

The model has a learning rate of 3e-4 and a batch size of 14 (because I trained the model using colab's GPU with a maximum RAM of 16, a batch size of more than 14 in the test would result in memory overflow). The model was trained 3000 times and the final loss was 0.11.

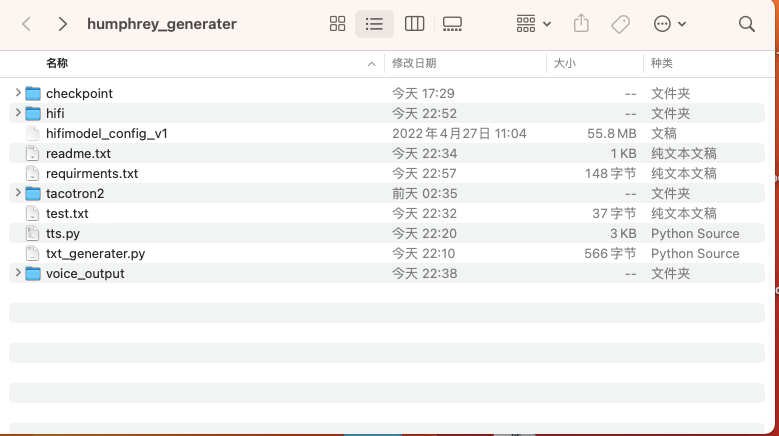
2.3 HiFi-GAN

HiFi-GAN is a high-quality generative adversarial network model for speech synthesis, whose main function is to transform melodic spectrograms into raw audio waveforms. [(Kong, Kim and Bae, 2020)](https://www.zotero.org/google-docs/?VWBfNm).It uses generative adversarial networks (GANs) to achieve high-quality speech synthesis. A GAN consists of two main parts: a generator and a discriminator. The role of the generator is to generate output samples, while the discriminator's duty is to distinguish the generated samples from the real ones.

Our model uses HiFi-GAN as a vocoder for a text-to-speech system to convert the results we generate using the Tacotron2 model into .wav files. Since we only used it to take care of the conversion, once we did not do any training on it.

3 Introduce to project and setup

The project file is a .zip file named humphrey\_generater.zip. Which contains 10 files shown below.



Where checkpoint holds the retrained model of GPT-2, the “humphery-300” file in tacotron2 is our trained sound model, and the original version of the model is in hifi, which we use to generate the audio. “txt\_generater.py” is the python script used to generate the text, and “tts.py” is the text-to-speech script.

First, please unzip the package to your target location. Please note that since the python script relies on relative path acquisition models as well as “HIFI-GAN” and tacotron2, please do not change the location of this hifi and tacotron2.

Readme file:

All code is running on a mac machine with an m1 chip, and has not been tested for applicability to other operating systems.

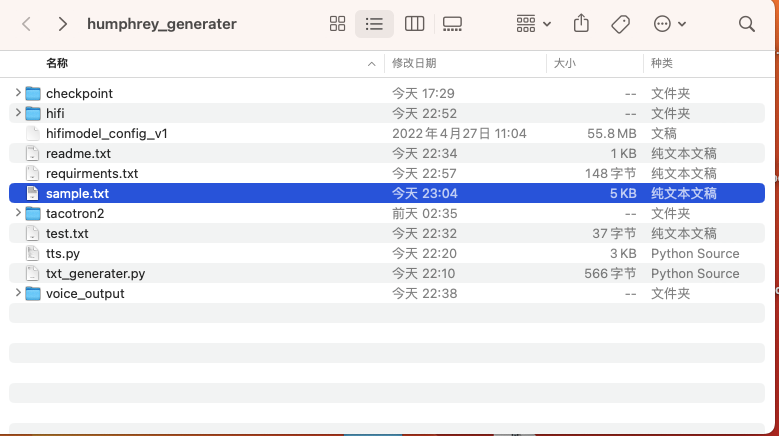
To install all the dependencies in requirments.txt, you can run the command:

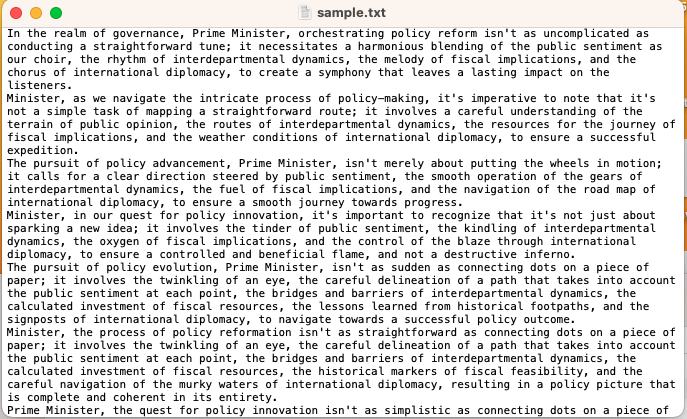
*pip install -r requirments.txt*

Run the txt\_generater.py file to generate the sentences. You can get the sample text file with the command: *python txt\_generater.py* with the name "sample.txt". You can also specify the name of the output text file with *python txt\_generater.py -n your name*. The output text will be kept in the current directory.

Run tts.py to convert the sentences to speech. You can generate a sample voice with the command *python tts.py* (which reads "yes, prime minister."). You can type the string you want to use directly with the -t command like: *python tts.py -t "yes, prime minister”*. You can also specify a text file with the -f command to make the program generate speech by line, e.g. *python tts.py -f "test.txt"*. The generated .wav file will be stored in the voice\_output folder.

When you successfully execute python txt\_generater.py, the script will generate sample.txt in the current path. The content inside is the words of Humphrey style generated by the GPT-2 retraining model. Shown below:





After you successfully generated the text, you can run *python tts.py -f sample.txt*  to generate the voice file.

4.System Creativity

The evaluation for creativity was limited to texts generated by the retrained GPT-2 model and did not involve sound cloning.

For the creativity of the system, we would like to assess the following aspects:fluency, flexibility, specificity, Similarity.

For fluency, GPT-2 seems to lag behind GPT-3 in terms of generation speed, but the reason may be that I use the CPU to generate the text. But even if it is not as fast as GPT-3, our model can still generate a large amount of text in a short time, which is very good.

The flexibility of the model was very unsatisfactory in the first few versions, and it generated sentences with a uniform template. However, after the adjustment of the training parameters and the improvement of the training data, its flexibility has improved significantly. But although our model is able to generate more types of sentences than the original version, it is still clear that the sentences have a similar logic, rather than a very diffuse thinking as in the original play.

For the specificity, this assessment is very interesting. Since our aim is to generate nonsense, the lower the score of this evaluation, the more successful our model becomes.The words generated by the model are all nonsensical nonsense, as in the following sentence (generated by the model):

“Minister, as we step onto the stage of governance, it is important to remember that the act of policy creation and implementation is not merely about reading lines from a script, but rather about giving a compelling performance that takes into account the audience's reaction, the delicate interplay among the ensemble of departmental actors, the budgetary constraints of the production, the historical context of the play, and the review of our performance on the international stage, all in order to ensure a standing ovation for our governmental operations.”

“It's incumbent upon us, Prime Minister, to grasp the reality that the process of policy formulation and execution in our democracy isn't simply an act of ideation and implementation, but an elaborate ballet requiring the precise coordination of a myriad of elements such as the whims and sentiments of the public, the labyrinthine inner workings of interdepartmental relations, the careful consideration of the potential economic implications, the conscientious study of historical precedents, and a thorough understanding of our international obligations, to ensure that each policy is balanced and well-suited to the needs of the present while keeping an eye on the future.”

For the Similarity( compare with “yes, minister.” and “yes, prime minister”), the sentences we generated basically include minister or prime minister, and the discourse in the sentences basically revolves around government and politics, so the model scores well on this point. But in the original play, Sir Humphrey’s words are more complex in structure, which our model cannot achieve very well.

5.Challenge

In this chapter, we will discuss the problems encountered in training text models and sound clones, respectively.

5.1 GPT-2

The process of retraining GPT-2 has been generally smooth for us. One of the challenges we encountered was that the text generated using GPT-4 had a very homogeneous structure, which also affected the results generated by the model. After noticing this problem, I tried to have GPT generate texts with different structures, but the results were not satisfactory. Finally, by adding some lines that appeared in the original play, our model succeeded in generating sentences with multiple patterns.

5.2 voice clone

We had a lot of challenges with the sound cloning problem and actually spent 90% of our working time.

The first challenge we encountered was environmental. Although I have a mac and a windows machine with a graphics card, one of them does not have a cpu for training and the other is not compatible with the new tensorflow framework. After about a day of wasted time, I opted to use Google's colab remote for training.

The second challenge is the quality of the training data. To test whether the model can be successfully trained, I record 25 voices of myself reading English sentences and use them as training data. Surprisingly, despite using only 25 short audio recordings, the model generated audio that perfectly cloned my voice and even my accent.

But when I started training with the original voice from "yes,minister", the resulting audio was so bad that you could clearly hear the noisy background sounds. We believe there are two main reasons for this: first, the TV show was broadcast in the 1980s, and even though the audio I selected had been restored by the AI, it didn't help with the training. Second, the recording of the show itself included a noisy environment, and for better viewing, the show's crew recorded laughter as well, which made it extremely difficult to prepare our data.

To solve the problem, I initially chose to increase the number of audio bars to allow for better sound cloning. As the training data increased, it became increasingly apparent that the background noise problem was not alleviated, although I could see that the sound cloning was getting better and better. And as the number of data bars increased, the model converged slower and slower, and at a later stage we trained a sound cloning model that cost tens or even hundreds of times more than training GPT-2, which led me to almost use up the available resources on colab.

At a later stage of the experiments on the problem, I tried to noise reduce the data before they were trained. I conducted experiments using a small dataset after the noise reduction process and found that the background noise was indeed successfully removed. Unfortunately, I was unable to use the processed large dataset for training because the previous training took up the running time of Cloab.

The third problem is that the model has a major problem in handling long sentences. When the length of the sentence is too long, the model is prone to decoding failure, a phenomenon that seems difficult to describe in words, but is very obvious if you run the program I provided, especially when feeding in the very long sentences we generated. The solution to this problem is to increase the amount of data. After comparing various versions (50 training data, 98 training data, 146 training data, and 182 training data), we found that the sentences that the model can handle do get longer as the training data increases. Another possible reason is that most of the sentences we trained were not long enough. Due to the large amount of laughter in the program, it was difficult for us to find a long sentence without the interference of laughter, which may also cause the model to have difficulty in handling long sentences.

6.Conclution

Although our project encountered many challenges, we did accomplish our intended goal (the end result was not entirely satisfactory) of generating Sir Humphrey's words and reading them in his voice. In achieving our purpose, we used three different AI models and retrained two of them, which gave us a deeper understanding of how to use big models.We also found possible solutions to the problems we encountered during the production of the project through several attempts, which can be used to better improve our project in the future.

Bibliography

[Kong, J., Kim, J. and Bae, J. (2020) ‘HiFi-GAN: Generative Adversarial Networks for Efficient and High Fidelity Speech Synthesis’. arXiv. Available at: https://doi.org/10.48550/arXiv.2010.05646.](https://www.zotero.org/google-docs/?a5jFEs)

[Shen, J. *et al.* (2018) ‘Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions’. arXiv. Available at: https://doi.org/10.48550/arXiv.1712.05884.](https://www.zotero.org/google-docs/?a5jFEs)

[Solaiman, I. *et al.* (2019) ‘Release Strategies and the Social Impacts of Language Models’. arXiv. Available at:](https://www.zotero.org/google-docs/?a5jFEs) <https://doi.org/10.48550/arXiv.1908.09203>[.](https://www.zotero.org/google-docs/?a5jFEs)

<https://github.com/jik876/hifi-gan>

<https://github.com/minimaxir/gpt-2-simple>

<https://github.com/justinjohn0306/FakeYou-Tacotron2-Notebook>

https://github.com/NVIDIA/tacotron2

https://github.com/openai/gpt-2