

The Chinese University of Hong Kong
Department of Computer Science and Engineering
Final Year Project Report (Semester 1)

Machine Learning assisted
Cantopop Lyric Composition

LYU2101

Supervisor:

Prof. Michael R. Lyu

Author:

Tang Ka Lok, 1155125745

Abstract

Cantopop is very special compared to other types of pop music like English pop, Mandarin pop, J-pop, etc. It's because the lyrics of Cantopop need to follow the tone-melody matching mechanism which make producing Cantopop unique and difficult.

Due to the recent state-of-the-art (SOTA) performance in the field of Natural Language Processing (NLP) using different machine learning techniques. This project aims at proposing a tone-based lyrics generation approach that apply machine learning to train a model which take tone as the input and generate lyrics matching the tone input as the output in order to assist Cantopop lyrics composition under the limitation of tone-melody matching mechanism. Bart model will be constructed to implement the tone-based lyrics generation and a web application will be developed for public to use to generate the lyrics.

Acknowledgements

I would like to express my gratitude to my supervisor and adviser, Professor Michael R. Lyu and Mr. WANG, Wenxuan for offering valuable guidance and opinions. I won't be able to finish the project without their generous help.

Table of Contents

Abstract	2
Acknowledgements.....	3
1. Introduction	6
1.1 Overview	6
1.2 Motivation.....	7
1.3 Objective	7
2. Background	8
2.1 Nature of Cantonese and Cantopop	8
2.2 Machine Learning and Natural Language Processing	11
2.3 Model Architecture	12
2.3.1 Recurrent Neural Network (RNN)	12
2.3.2 Long Short Term Memory network (LSTM)	14
2.3.3 Transformer.....	15
2.3.4 Bert.....	19
2.3.5 GPT	20
2.3.6 Bart.....	21
3. Related Work.....	22
3.1 Word/Sentence-prediction-based Lyrics Generation	22
3.2 Melody-based Lyrics Generation	24
4. Methodology.....	26
4.1 Tone-based Lyrics Generation	26
4.2 Training from Scratch, Pre-training and Fine-tuning.....	28
4.3 Data Preparation	29
4.4 First Stage: GPT-2	30
4.5 Second Stage: Bart	31
4.6 Model Evaluation Metrics	33
4.6.1 BLEU	33
4.6.2 Perplexity	34
4.6.3 Tone Accuracy	35
5. Experiments	36
5.1 Data Crawling	36
5.1.1 Traditional Chinese Corpus data	36
5.1.2 Lyrics Data	37
5.2 Data Preprocessing	38
5.2.1 Data Cleaning	38
5.2.2 Tone2Text dataset building	42

5.2.3 Tokenization	43
5.3 GPT2 Models	44
5.3.1 Structure	44
5.3.2 Statistics	46
5.4 GPT-2 Evaluation	48
5.4.1 BLEU	48
5.4.2 Perplexity	48
5.4.3 Sample.....	49
5.4.4 Comparison	51
5.5 Bart Model	52
5.5.1 Structure	52
5.5.2 Statistics	54
5.6 Bart Evaluation	56
5.6.1 BLEU	56
5.6.2 Perplexity	56
5.6.3 Tone Accuracy	57
5.6.4 Sample.....	58
5.6.5 Models Comparison	60
6. Tone2Cantopop	61
6.1 Lyrics Generation	61
6.2 Text-to-speech.....	63
6.3 Tone Comparison	64
7. Limitation	65
7.1 Size of dataset	65
7.2 Multiple possible tones of a Chinese character.....	66
8. Conclusion.....	67
9. Possible Future Development	68
9.1 Model improvement	68
9.1.1 Theme/Title based lyrics generation	68
9.1.2 Lyrics generation with partly finished lyrics.....	68
9.1.3 Model Configuration Adjustment	68
9.2 App Improvement	69
9.2.1 Account system	69
9.2.2 Rating system	69
10. Reference	70
Appendix	73

1. Introduction

1.1 Overview

In recent years, many tasks that was considered as impossible for machine to achieve are finally overcame including but not limited to beating the world's best go player, having a bot who talks like a human or even looks like a human, etc. One important reason for such rapid improvement is definitely machine learning. Machine learning gives the power to machine to learns and acts like a human or even overtakes human.

Creative industries have been considered another field that is impossible for machine overtaking human as many people think that machine doesn't have the creativity as human. However, due to the increasing power of machine learning, the performance of machine in this field gets better and better as well. It's hard to expect how far can it reach. It is believed that the day the machine can beat human even in creative industries will finally come.

Applying machine learning into NLP tasks is a popular topic recently. This gives machine the ability of understanding human language which help machine performs better in the field of creative industries. Lyrics composition is one of the NLP tasks as well as one important creative industry. This project focuses on utilizing the power of machine learning to generate meaningful and high-quality Cantopop lyrics under the limitation of Cantonese feature, the tonal system. This report demonstrates the process of this project for the 1st semester and this chapter gives a brief introduction to the topic.

1.2 Motivation

Cantopop refers to the songs and music videos created by the Hong Kong music industry. Back to 70s-90s, Cantopop is extremely famous among the worlds. However, since 00s, Cantopop industry starts to decline and the voice about the industry is dead keep rising. Cantopop gradually lost its influence in the mighty torrent of history [20]. I grew up with Cantopop and hence I become a Cantopop lover due to my culture background. Cantopop is the collective memory of all Hongkongers and it's our valuable local culture. Therefore, that's the reason I want to make contribution to Cantopop.

Lyrics composition is always the most difficult part in Cantopop creation. However, due to the increasing power of machine learning in the field of natural language processing (NLP), we can apply machine learning to different kinds of NLP tasks including text-generation which can produce a better and better result in recent years. This encourages me to use a machine learning approach to do the Cantopop Lyrics Composition to increase both quality and quantity of Cantopop lyrics and also lower the barrier of entering the field of writing Cantopop lyrics.

Therefore, I want to utilize the power of machine learning to make contribution to the culture of Cantopop because of my interest on Cantopop and the increasing power of machine learning on handling NLP tasks.

1.3 Objective

The final goal of this final year project is making use of the machine learning technique to build a model to assist the Cantopop lyrics composition tasks. As Cantopop lyrics composition is extremely difficult with the limitation of melody matching the tone of lyrics. Therefore, this project aims at developing a system that can generate Cantopop lyrics under the tone limitation. Specifically, when user input the theme/name of the song and the tone of the lyrics, the system can generate some sample lyrics to user. In 1st semester, the focus was collecting data and finding a best way to achieve the goal of this project.

2. Background

2.1 Nature of Cantonese and Cantopop

Cantonese is a kind of tone languages which means that there is a specific tone representing a unique pitch associated to a word. A tone language uses pitch to distinguish between different words and meanings [14]. When the tone of the word is different, it may represent a totally different word even the syllable is the same. In the system of Cantonese, there is a saying, “nine sounds and six tones” (九聲六調). We can loosely treat it as six distinct pitch contours in Cantonese [17] which means that there can be 6 (or even 9 if treat it strictly) different meaning with the same syllable. Figure 2.1 gives a description of six tones, explaining the pitch contours of them.

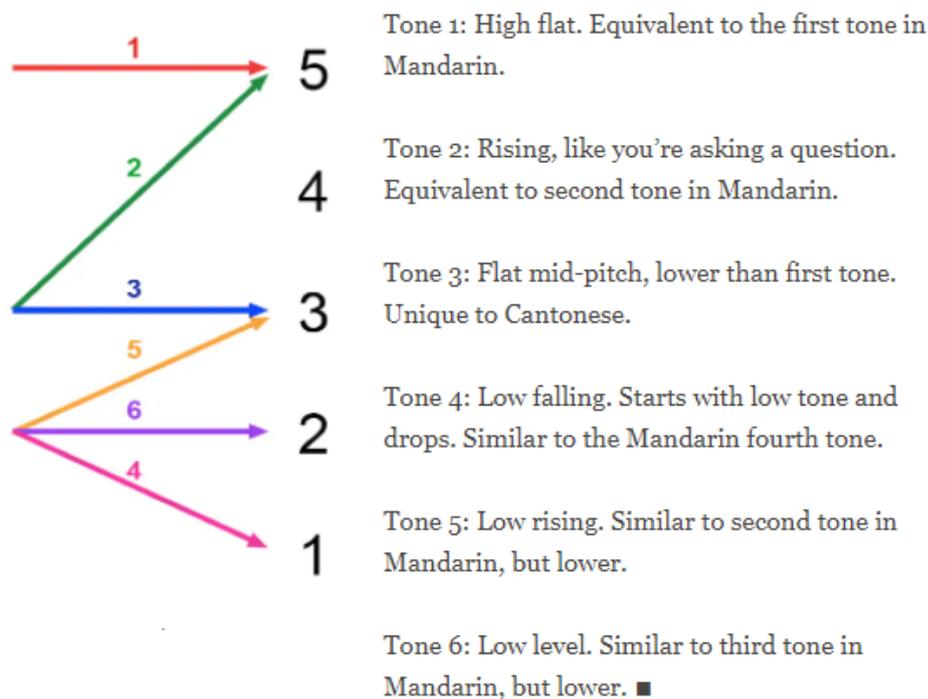


Figure 2.1: General Description of six Cantonese tones

Source: [17]

Figure 2.2 gives the example how the same syllable with different tones gives different word and meaning. Each word is combining the syllable ‘fan’ and tone from 1 to 6. This kind of representation is called Jyutping representation.

fan1	fan2	fan3	fan4	fan5	fan6
昏	粉	訓	焚	奮	份

Figure 2.2: 6 words having same syllable with 6 different tones

One important feature of Cantopop compared to other types of music like Mandarin pop and English pop is the association of tonal system of Cantonese which is called tone-melody matching mechanism [25]. Therefore, it's extremely important to matching the lyrics to the pitch of the song. If the pitch of the song doesn't match the tone of lyrics. The song is basically not understandable. Someone may ask why there isn't such limitation in Mandarin pop while Mandarin is also a tone language. The reason is that the nature of 4 different tones in Mandarin is difficult to construct a complete music scale while the tone in Cantonese is able to construct a Tetratonic Scale (4 notes) [21]. Therefore, if the lyrics can't match the pitch of the song, it's basically not understandable. Because of nature of Cantonese, it's extremely difficult to do lyrics composition in Cantopop. Below figure shows an example illustrating the situation when the tone not matching the melody:

This is a lyrics line from a Hong Kong famous kid's song with Jyutping representation:

Original Lyrics	
	ngo5 mun4 si6 faai3 lok6 dik1 hou2 ji4 tung4 ngo5 mun4 tin1 tin1 jat1 hei2 go1 coeng3
	我 們 是 快 樂 的 好 兒 童 我 們 天 天 一 起 歌 唱
	(We are happy good children. We are singing together every day)
Music Notation	
Lyrics that match the melody	ngo2 mun5 si6 faai3 lok3 dik6 hou2 ji5 tung3 ngo2 mun6 tin1 tin1 jat1 hei3 go3 coeng1
	鶻 滿 是 快 烙 滴 好 耳 痛 鶻 閱 天 天 一 戲 個 窗
	(not translatable)

Figure 2.3 Example of fitting lyrics with unmatched tone into the melody

From Figure 2.3, we can see that if we force the lyrics to match melody, the lyrics will become some other words with the same syllable but different tone which can't express the origin meaning of the lyrics at all.

Due to such nature of Cantonese, most of Cantopop is created based on the music-first lyrics-second order. Because creating melody based on the lyrics under the limitation of tone-melody matching mechanism greatly restricts the variation of the music. Here is a general step for a Cantopop lyricist to write lyrics:

1. Convert the melody to the tones of Cantonese based on the pitch of the music notes.
2. Fill in the lyrics that match the tones.

Figure 2.4 shows an example how a Cantopop lyric is writing with the above steps.

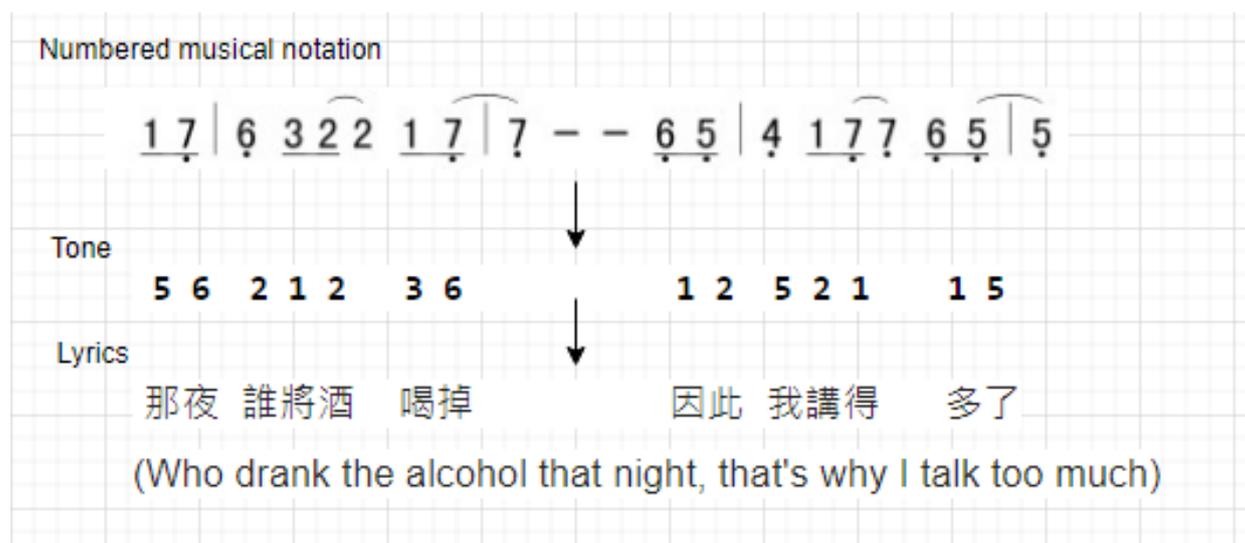


Figure 2.4 Example of the steps to write Cantopop lyrics

This project mainly focuses on the second step to help lyricist fill in lyrics that match the tones.

2.2 Machine Learning and Natural Language Processing

Both Machine Learning and Natural Language Processing (NLP) are subfields in Artificial Intelligence (AI). NLP aims at helping machine to understand and analyze human languages. NLP tasks include but not limit to [15]:

- 1) Machine Translation
- 2) Text Summarization
- 3) Auto-Predict
- 4) Natural Language Generation

Applying machine learning is a common approach in NLP. Particularly in recent years, machine learning plays a very important role in the field of NLP. Given the power of deep learning that largely increase the learning ability of machine, applying deep learning techniques greatly rapid the development of NLP as well.

Here introduces several machine learning approaches that play an important role in NLP:

1. Recurrent Neural Network
2. Long short term memory network (LSTM)
3. Transformer

NLP is applied to many things in our daily life nowadays. Some well-known application would be Siri, google translate, etc.

This project aims at utilizing and exploring the ability of the machine learning on Cantopop Lyrics Composition which belongs to the field of NLP.

2.3 Model Architecture

Several model architectures related to this project are introduced below. They are some popular architectures that perform very well in the field of NLP.

2.3.1 Recurrent Neural Network (RNN)

RNN is a kind of neural networks that is good at handling sequence. Text is also a kind of sequence data and hence, RNN is a popular way for doing NLP tasks. A common problem of other kinds of neural networks is that these types of networks take fixed-size inputs and give out fixed-size outputs [16]. Consider a simple NLP task like text summarization, the model generates summary given a articles input. The model should accept articles with different length and produce corresponding summaries with different length as well. Therefore, those networks with fixed-size inputs and outputs are not suitable for handling NLP tasks. RNN is the network that can accept different length of inputs and generate different length of outputs. Another issue of traditional neural networks handling sequence data is that they can't "remember" the information of previous input. Only current information is used to do future prediction for traditional neural network, and it's dumped immediate after it's used. RNN can track the information of previous inputs because the information is keep passing to next layers inside the networks [19].

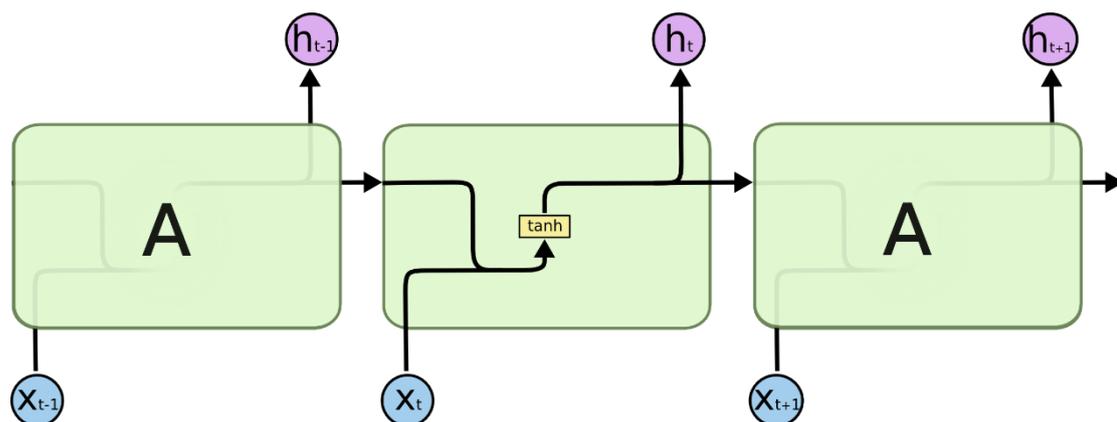


Figure 2.5 RNN Architecture

Source: [28]

Above figure shows a basic architecture and working principle of an RNN. The architecture is relatively simple, whole RNN is chained with many simple modules

which contain a simple layer responsible for calculating the state of current input. For a sequence data, RNN will read from the start of sequence, keep updating the cell state given the processed input partition and pass the information as vector sequence to the next module. For example, when an RNN is processing a sentence with 10 words and it is reading the 5th word, the RNN will update the cell state given the information of previous 4 words. Consider a person reading a sentence, he will keep processing the sentence word by word from left to right, keep updating his understanding till the end of the sentence. The working principal of RNN is pretty much the same as a human reading a sequence data. That is one reason RNN performs that well in the field of NLP.

Although remembering previous information is one big advantage of RNN, there is also one big limitation for RNN. RNN can only remember short-term dependencies. Once the sequence is long, the networks may not be able to depend on the information that's far away from current position as it's already lost through time [19]. Consider reading an article with three paragraphs, although the contents of the third paragraph may be depending on the contents of the first paragraph, we may not be able to remember the information of the first paragraph and hence, we have to read it again to gain the information. It's one big constraint of RNN that it can't remember long-term dependencies.

2.3.2 Long Short Term Memory network (LSTM)

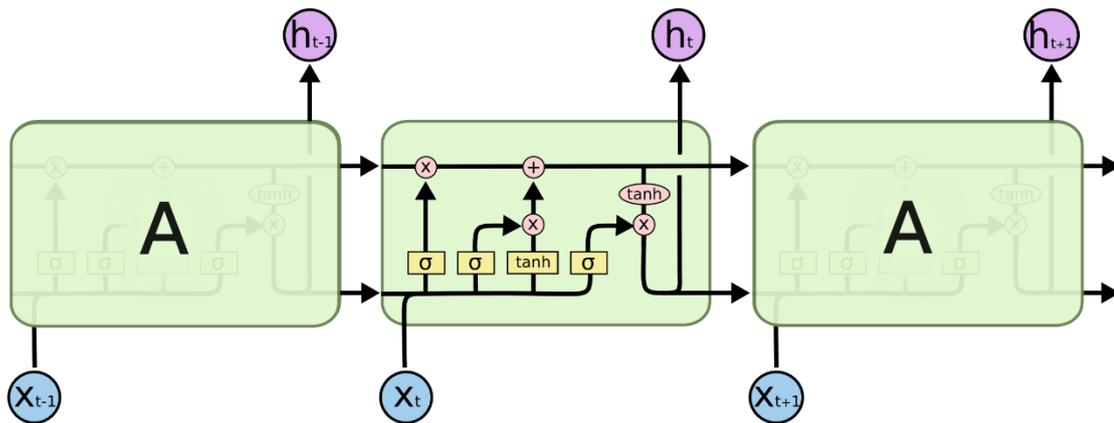


Figure 2.6 LSTM Architecture

Source: [28]

LSTM is a modified version of RNN to deal with the problem of poor ability of RNN remembering long-term dependency. The above figure shows a modified architecture compared to RNN. LSTM is still chained modules by modules but within each module, things get much complex compared to RNN module. To put things simple, what each module of a LSTM doing is that by setting different gates, the network is able to decide what information to be kept, removed, updated and outputted [28]. We can see that there are four different layers colored in yellow within a module. They are gate layers to decide the flow of the information, followed by corresponding operations colored in pink. This is the main idea how LSTM handling the information in order to achieve the function of remembering long term dependencies.

2.3.3 Transformer

Although there are some variations of RNN to solve the limitation of RNN, there are still some big limitations that are hard to solve due to the nature RNN structure.

Parallelization has always been one big problem of RNN. As describe above, as RNN read the sequence data in a sequential manner just like human reading a sentence, so the way of RNN handling the data must be linear, from the start of sequence to the end of sequence. Due to the disability on parallelization, the computation speed is slow with RNN structure [23].

Also, even LSTM seems to be able solve the long term dependencies problem, but still, the data is processed in a sequential manner. As a result, once the sequence is getting longer and longer, and the information is passed layer by layer and being processed with different operation, the information will be lost eventually [23].

However, there is a very important breakthrough to solve the above problems of RNN. A new mechanism called Attention mechanism is first proposed by Bahdanau et al. in 2015 [16]. Put it in simple, attention mechanism is calculating alignment score between input and output by passing matrix of vector of the hidden states instead of just passing the last single vector compressed all the information in traditional RNN model [29]. However, attention mechanism is applied only between encoder and decoder initially which still keeping the RNN structure within encoder and decoder. Therefore, in 2017, a model structure called Transformer is proposed in the paper “Attention is all you need” by Google which utilizes the power of attention mechanism by applying it to the encoder and decoder as well and totally gives up RNN structure [5]. This is called self-attention mechanism.

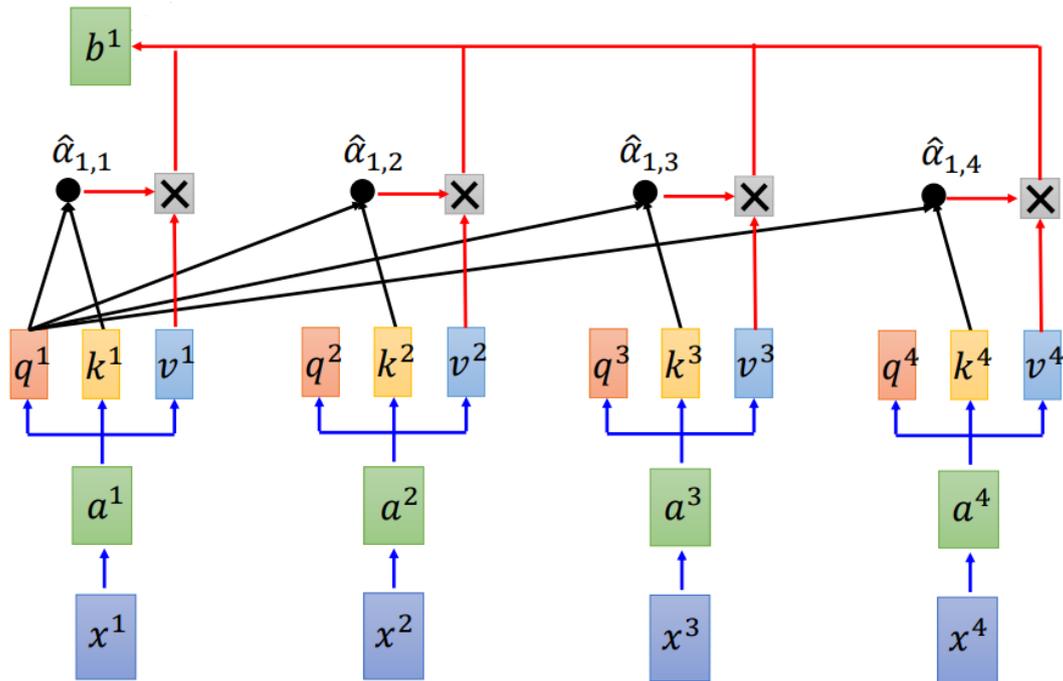


Figure 2.7 Self-attention mechanism working principle

Source: [26]

The above figure generally describes the working principle of self-attention mechanism. For a sequence data, it's not handled in a particular order. Instead, all input is handled at the same time which achieve parallelization that RNN unable to achieve. Each input x^i is converted three different vectors. They are query vector q^i , key vector k^i , value vector v^i [26]. These three vectors are used to calculate the attention scores of each input to decide how much attention should an input put to other inputs.

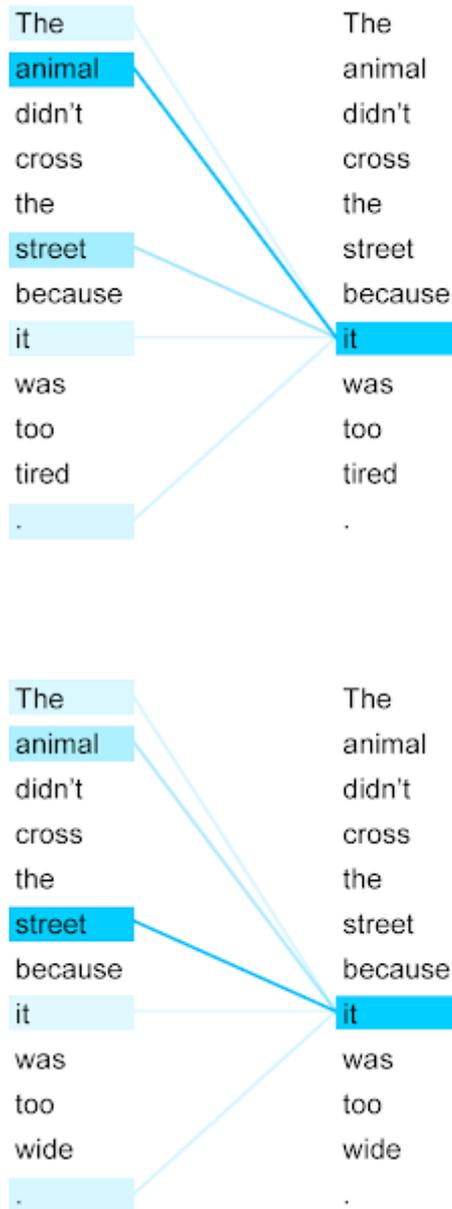


Figure 2.8 Self-attention mechanism visualization

Source: [27]

Above figure is a simple example illustrating the self-attention mechanism. The line connecting between two words meaning the attention of a word putting on another word and the darker color meaning a higher attention scores. We have two sentences here. The animal didn't cross the street because it was too tired and the animal didn't cross the street because it was too wide. These two sentences only differ in one word, tired and wide, but they are illustrating different meanings and hence, "it" in two sentences are actually referring to different words. In this first sentence, "it" refers to

animal as the sentence is illustrating the animal was too tired. In the second sentence, “it” refers to street as the sentence is illustrating the street is too wide. The power of self-attention mechanism is that it can correctly put the attention to the word that it actually refers to.

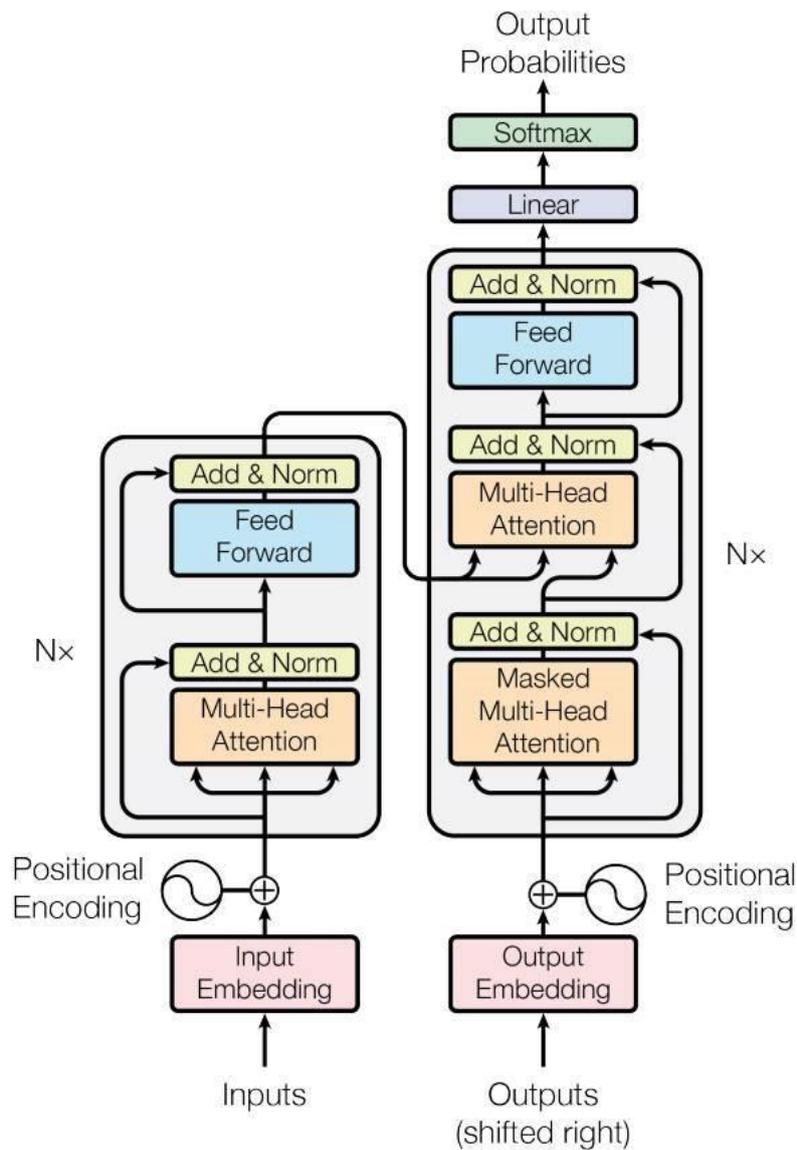


Figure 2.9 Transformer Architecture

Source: [5]

The above figure shows the model structure of Transformer which apply self-attention mechanism. The left part is the encoder, and the right part is the decoder. For the encoder part, each encoder layer consists of two parts which are self-

attention layer and feed-forward neural network which are responsible for calculating and outputting the attention score of the input sequence. In each decoder layer, it consists of one more part called masked attention layer which is responsible for calculating the attention score of the generated output. Finally, the probabilities of next output will be calculated based on the attention outputted [24].

2.3.4 Bert

BERT stands for Bidirectional Encoder Representations from Transformers. We can know it from the name that Bert is a variation of Transformer. Bert dumps the decoder part and construct the model using Bidirectional Encoder [7]. In such way, we can train the model in an unsupervised way to learn the representation while transformer still requires labeled dataset to train the model in a supervised way. Therefore, Bert is considered a language model while transformer is a kind of sequence-to-sequence model.

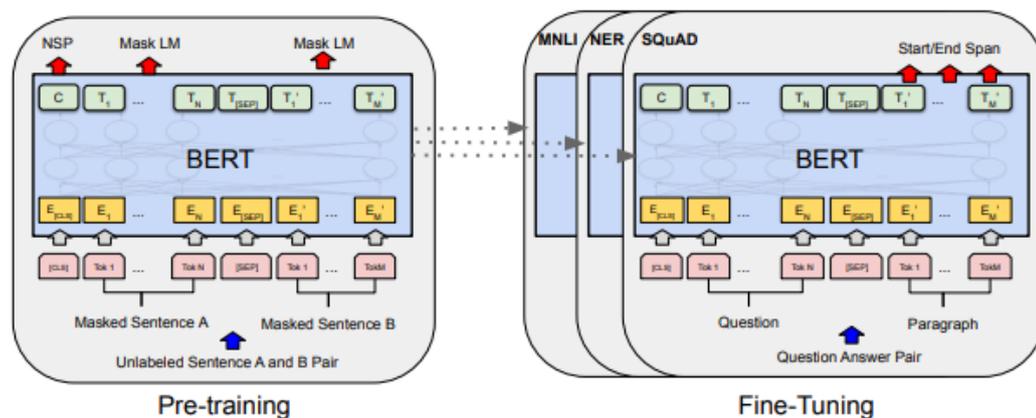


Figure 2.10 Bert working philosophy

Source: [7]

The above figure illustrates the working philosophy of Bert which utilizes the power of pre-training and fine-tuning. The true power of Bert is that a pre-trained Bert model which well-learn the representation of the data, can be easily fine-tuned for different NLP task with only adjustment on the structure, which is mainly the adjustment on the final output layer to fit the output requirements of different tasks.

2.3.5 GPT

GPT stands for Generative Pre-trained Transformer. We can know it from the name that GPT is another variation of Transformer which also focus on utilizing pre-training and fine-tuning. The different between Bert and GPT is that Bert is an encoder-only transformer-based model while GPT is a decoder-only transformer-based model. Same as Bert, GPT is considered a language model as it can be trained with unlabeled data in an unsupervised way. Consider the original transformer, the decoder of transformer is used to generate next output based on the previous input. Therefore, the working principle of GPT is an auto-regressive model which is actually doing next word prediction i.e., a one direction model that predict the next output from left-to-right.

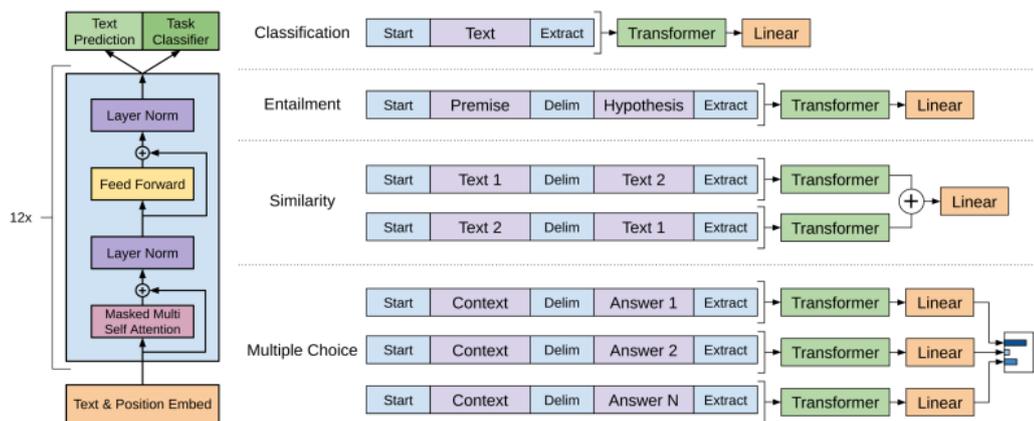


Figure 2.11 GPT Architecture

Source: [10]

The above figure illustrates the structure and working principle of GPT, we can see that the only different between GPT and the decoder part of Transformer is that one self-attention layer is taken off as there isn't input from encode anymore. Same as Bert, with a well-pretrained GPT model, it can be fine-tuned for different kind of tasks.

There are some improved versions of this model called GPT-2 and GPT-3 [11][12]. Both of them are well-pretrained with a huge amount of data and can be directly used for fine-tuning. GPT-2 model is already open to public while GPT-3 is not released yet.

2.3.6 Bart

Bart stands for Bidirectional and Auto-Regressive Transformers. We can know it from the name that Bart is also a variation of Transformer and one step further, it's also the combination of Bert and GPT. Combining a bidirectional encoder (Bert) and an autoregressive decoder (GPT) to form a sequence-to-sequence model which get back to the original structure of Transformer.

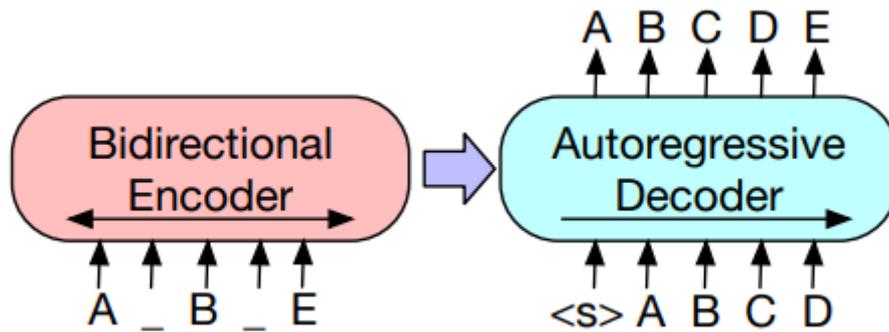


Figure 2.12 Bart Architecture

Source: [6]

Above figure illustrates the simple structure and the working principle of Bart. As Bert is an encoder only model, it lacks the ability of text generation. As GPT is a decoder only model, it lacks the ability of learn the bidirectional contextual information [6]. Combining the advantages of both architectures, Bart achieves a state-of-the-art (SOTA) performance in the field of NLP.

3. Related Work

There is much research and implementation about generating music lyrics.

Although no one research can directly apply into the mechanism of Cantopop lyrics composition, we can take the research applied into other type of music like English pop and Mandarin pop as reference to develop a unique model specially fit for the Cantopop lyrics composition.

3.1 Word/Sentence-prediction-based Lyrics Generation

The first approach of lyrics generation is to build a language model for pure text generation. This approach mainly focuses on the contextual information of the lyrics which aims at generating fluent and meaningful lyrics.

There are many ways to build such lyrics generation model and I am going to introduce two of them. The first one is the LSTM-based approach and the second one is the GPT-2-based approach. Both approaches give excellent result on text generation task and thus apply to lyrics generation task.

LSTM is a type of Recurrent Neural Network. It's good at handling sequence data which is a perfect match for text data. LSTM has the capability to learn long-term dependencies compared to pure RNN which make the quality of lyrics generated much higher as such architecture is able to take the contextual information of the lyrics into consideration and generate meaningful and fluent lyrics given an input sentence. One related work is a LSTM-Based model that can generate lyrics given a genre and stating lyrics sentence done by Harrison Gill, Danie (Taesoo) Lee and Nick Marwell [8].

GPT-2 is Transformer-based structure utilizing the power of self-attention mechanism. GPT-2 released by OpenAI is an extremely huge pretrained model which trained on 40GB of high-quality content that already able to generate human-like text. Given its huge amount of parameter (1.5B parameters), fine-tuning it specific to lyrics generation task to generate high quality of lyrics would be much

easier [10]. A related work is a GPT-2 model that can generate lyrics with specific genre done by Lau, Wesley [4]. However, the origin GPT-2 model mainly pretrained on English data and hence the origin pretrained model released by OpenAI can't be directly used for this project.

The above approaches focus only on pure text generation. However, in this project, the focus is to generate lyrics under the limitation of tone which is extracted from the song melody. Therefore, this is not a suitable approach to deal with the Cantopop lyrics generation.

3.2 Melody-based Lyrics Generation

Another approach for lyrics generation is to build a sequence-to-sequence model that takes the melody as input and generate corresponding lyrics as output. This approach takes melody into consideration and aims at generating meaningful lyrics that can capture the relationship between melody and lyrics. There are several excellent works investigating such melody-to-lyrics generation tasks.

An RNN-based melody-conditioned language model to generate Japanese song was proposed by Kento Watanabe et al. [2]. It prepared a collection of melody-to-lyrics data for analyzing the relation between melody and lyrics and built a model for taking melody as input and aiming at generating fluent lyrics that fit the melody.

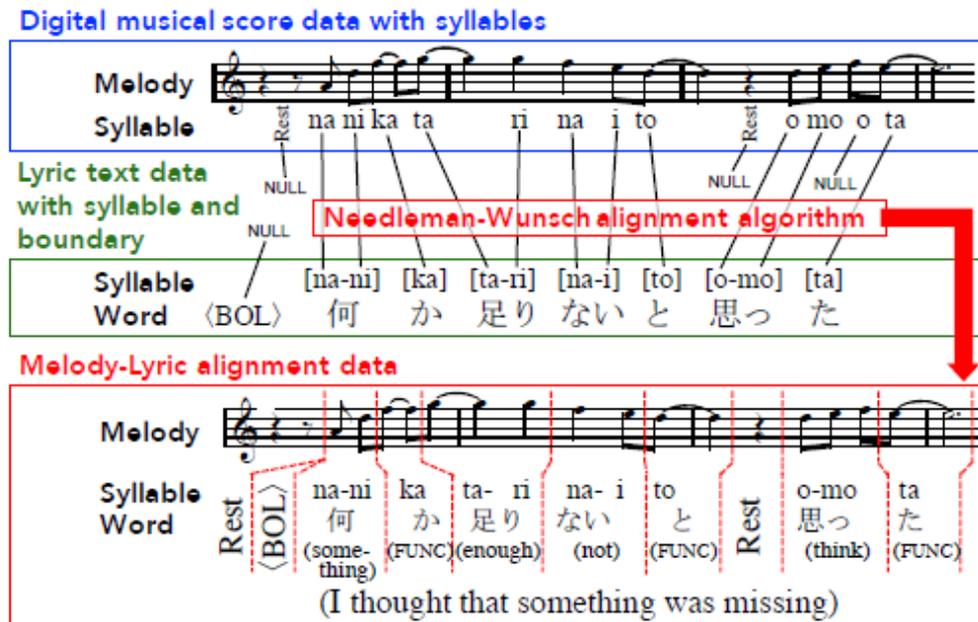


Figure 3.1 melody-to-lyrics data preparation

Source: [2]

The proposed approach also takes the pitch of the music notes into consideration which seems to be a perfect match for this project. As Cantopop lyrics highly relies on the pitch of notes that the tone of lyrics should perfectly match that pitch of melody due to the nature of Cantonese. However, one big constraint for applying this approach is that it's extremely hard to prepare a large amount of melody-to-lyrics data. Even in the Kento Watanabe et al.'s experiments, they are only using

1000 melody-to-lyrics data. Due to the limited resources in the field of Cantopop, it's even harder to obtain melody-to-lyrics data. Therefore, although such approach seems to be fitting this project, it's too difficult to implement such approach.

To deal with the problem of rare melody-to-lyrics data, another melody-based lyrics generation is a two-channel Seq2Seq generation model to generate Chinese lyrics proposed by Xu Lu et al. [3]. Instead of preparing melody-to-lyrics data, it represents the melody-to-lyrics data in the form of structural alignment between lyrics and melody which is much easier to obtain such kind of data. This kind of data mainly takes beat pattern of the melody into consideration to represent the melody part and hence, the model can learn how to generate lyrics given the pattern of melody.



Figure 3.1 melody-lyrics structural alignment

Source: [3]

However, this approach can't not be applied to this project. This approach didn't take the pitch of melody into consideration as Mandarin pop doesn't care about the relation between the pitch of melody and tone of word whereas this is extremely important in Cantopop due to the tone-melody matching mechanism.

4. Methodology

4.1 Tone-based Lyrics Generation

Given the problem when applying the forementioned approaches into Cantopop lyrics generation. Here is a new proposed approach specifying for Cantopop lyrics generation, Tone-based Lyrics Generation.

As mentioned in section Nature of Cantonese and Cantopop, there are two steps for a Cantopop lyricist to write lyrics:

1. Convert the melody to the tones of Cantonese based on the pitch of the music notes.
2. Fill in the lyrics that match the tones.

Converting the melody to the tones is a basic step in Cantopop lyrics composition. The hardest part is to fill in fluent and meaningful lyrics that can match the tones. Therefore, to deal with the tone-melody matching mechanism of Cantopop lyrics composition, we can actually skip the first step and let user to do the favor. This project focuses on building a model that can finish the second step.

One very big advantage of such approach is that the tone of word can actually be extracted from the lyrics. We don't need much effort to get the additional tones data unlike the melody-based approach which require great effort but resulting in getting only a few melody data. Therefore, having this approach can reduce the difficulties on getting related data for the model training while maintaining the nature of Cantopop lyrics composition.

Second advantage is that we can expand the training dataset from Cantopop lyrics data to all Chinese pop lyrics data. As all Chinese word are associated with a Jyutping representation in Cantonese, we can just extract the tone of the word to form a tone-to-text dataset, even if the lyrics is from a Mandarin song. In melody-based approach, melody and lyrics are directly related to each other and hence, melody-to-lyrics dataset that forms by Cantopop music must be used in order to

maintain the tone-melody matching mechanism. However, in Tone-based approach, as melody and lyrics are indirectly related to each other, and they aren't tied together. The model only needs to learn about the relation between tone and word and hence, all Chinese lyrics data can be taking as training data. Such approach can greatly increase the diversity of the model.

This project mainly involves of two stages:

1. Build a GPT-2 model (LM) which responsible for pure lyrics generation.
2. Build a Bart model (seq2seq model) which responsible for implementing the Tone-based Lyrics Generation approach.

The first stage focuses on building a LM imitating other lyrics generation projects on other types of music like English pop and Mandarin pop. The model takes a starting sentence as input and generate the remaining lyrics as output. As mentioned above, many existing lyrics generation model are Language Model or melody-based Sequence-to-Sequence model which can generate lyrics given a starting sentence or melody respectively. Although these approaches can't be directly applied to this project due to the nature of Cantopop, we can still take it as reference for this project. In the first stage of this project, we focus on training a GPT-2 model to generate high-quality lyrics.

Based on the well-trained Language Model, we can further build a Cantopop-specify model. This model can leverage the trained GPT-2 model to initialize the decoder part of the Bart model as the Bart model is going to be trained with tone-to-text data and the model may not be able to generate semantic coherence lyrics given that the tone data non-contextual [13].

After the implementation of these stage, the final model which can take tones as input should be able to fills meaningful lyrics that match the tones as output.

4.2 Training from Scratch, Pre-training and Fine-tuning

After setting up the approach to achieve the final goal of this project, we need to define our ways to obtain the best model implementing such approach. Here comes with two different training approaches: Training from Scratch, Pre-training and Fine-tuning. Training from Scratch means directly train the model using the target data. Pre-training and Fine-tuning mean we first pre-train a model using in-domain data and then we fine-tune it to our specific task using the target data [9]. We can compare the results generated by the models training using these two approaches. The two approaches would be applied to both models mentioned above. Generally speaking, fine-tuning a pre-trained model should give a better result. However, when pretraining with a large amount of corpus data, the data is quite different from lyrics data in term of the sentence pattern, word choice or even the punctuation. Therefore, we can compare the result training from scratch and fine-tuning from a pretrained model to see which approach can give a better result.

As this project involves two stages, there are several combinations of above approaches which can be applied into the forementioned two models which can help us explore the differences of the models applying different approaches. Here are 6 different training approaches to be applied to this project.

1. Pretrain a GPT-2 with traditional Chinese corpus data and fine-tune it with Cantopop lyrics data.
2. Train a GPT-2 from scratch with the lyrics data only.
3. Pretrain a Bart model with tone to traditional Chinese corpus data and fine-tune the model with tone2lyrics data.
4. Train a Bart model from scratch with the tone2lyrics data only.
5. Leverage the pre-trained GPT-2 as the decoder of the Bart model, pretrain the Bart model with tone to traditional Chinese corpus data and fine-tune the model with tone2lyrics data.
6. Leverage the pre-trained GPT-2 as the decoder of the Bart model and keep training the Bart model with tone2lyrics data.

4.3 Data Preparation

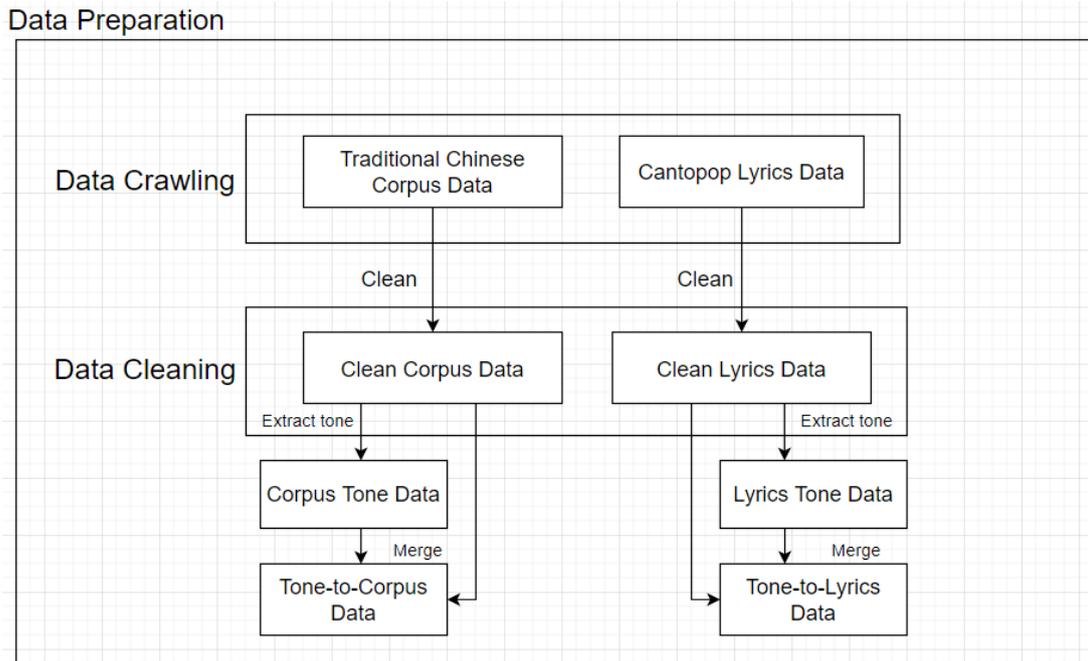


Figure 4.1 Flow chart of data preparation phase

The above diagram shows the flow of the data preparation stage. Data Preparation is always the first step of any Machine Learning Project. Based on the nature of the project, we need to prepare corresponding data to train the model. This is a project related to Cantopop so it's a must to prepare enough Cantopop lyrics data. Besides, we will compare the models with pretraining and without pretraining. Therefore, we also need to prepare large amount of corpus data to do the pretraining.

However, except for those already clean data available online, almost all data you crawl from Internet contains many unnecessary noises which are super messy and hence, proper data cleaning must be carried out in order to fit clean data to the model for training.

As mentioned above, the final goal of this project is taking tone of lyrics as input and outputting meaningful lyrics that match the tone. Therefore, we need to prepare tone-to-text data as well. We need to prepare two sets of tone-to-text data. They are tone-to-corpus data and tone-to-lyrics data that tone-to-corpus data is used for pretraining, and tone-to-lyrics data is used for fine-tuning/training from scratch.

4.4 First Stage: GPT-2

First Stage: GPT-2

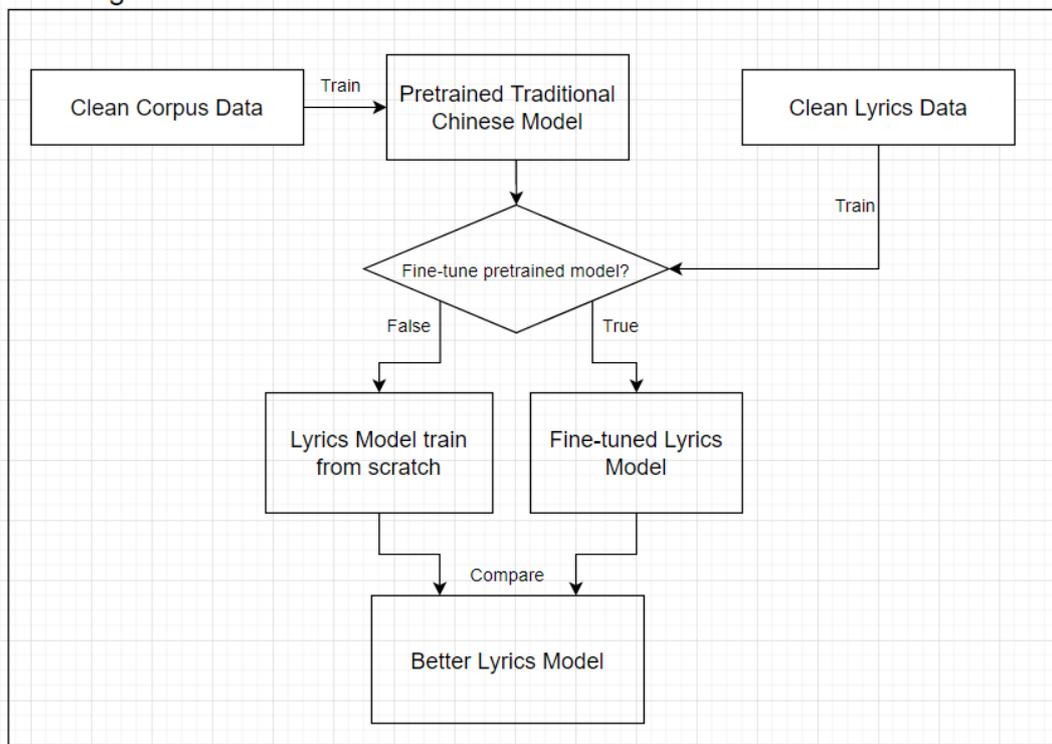


Figure 4.2 Flow chart of GPT-2 model phase

The above diagram shows the flow of first stage to train a GPT-2 lyrics model for generating high-quality lyrics given a starting sentence. Two different GPT-2 models are trained to be compared in order to get a better lyrics model. One model is fine-tuned with clean lyrics data using a pretrained Traditional Chinese Model which is trained with clean corpus data. Another model is just a model training from scratch with clean lyrics data without using any pretrained model.

Making use of the GPT-2 architecture, the final lyrics model should be able to generate lyrics that is fluent and meaningful and has lyrics-like structure,

4.5 Second Stage: Bart

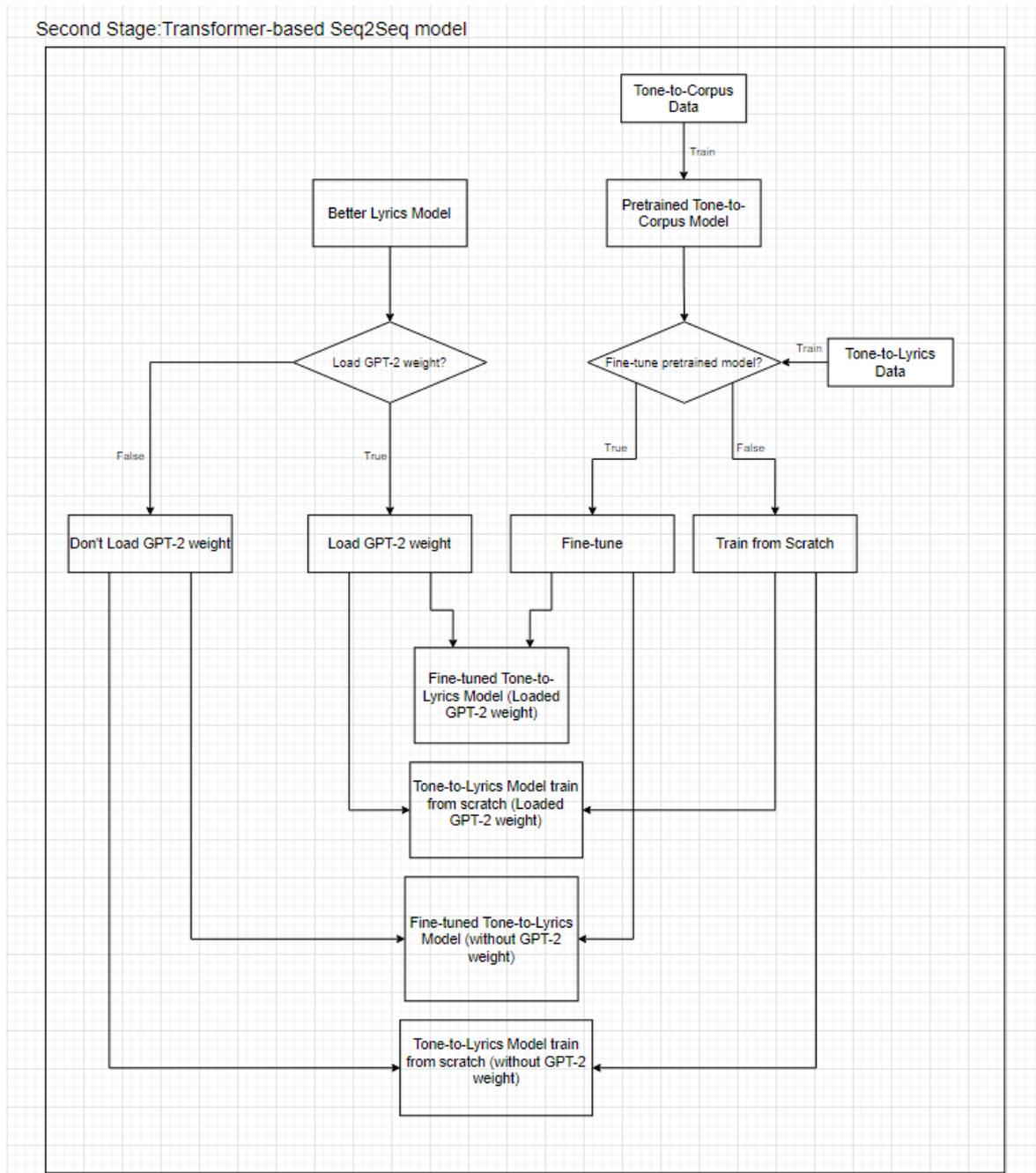


Figure 4.2 Flow chart of Bart model phase

The below diagram shows the flow of second stage to train a Bart model for tone2lyrics generation that taking tones as the input and outputting lyrics matching the tones. Four different Bart models are trained in this stage. Here are the differences between the four models.

1. Leverage the pre-trained GPT-2 as the decoder of the Bart model, pretrain the

- Bart model with tone to traditional Chinese corpus data and fine-tune the model with tone2lyrics data.
2. Leverage the pre-trained GPT-2 as the decoder of the Bart model and keep training the Bart model with tone2lyrics data.
 3. Pretrain a Bart model with tone to traditional Chinese corpus data and fine-tune the model with tone2lyrics data.
 4. Train a Bart model from scratch with the tone2lyrics data only.

After training 4 models with different ways, the last step for second stage is to compare the performance of these 4 models in order to obtain the best tone-to-lyrics model which should be able to generate high-quality lyrics which can exactly match the input tones.

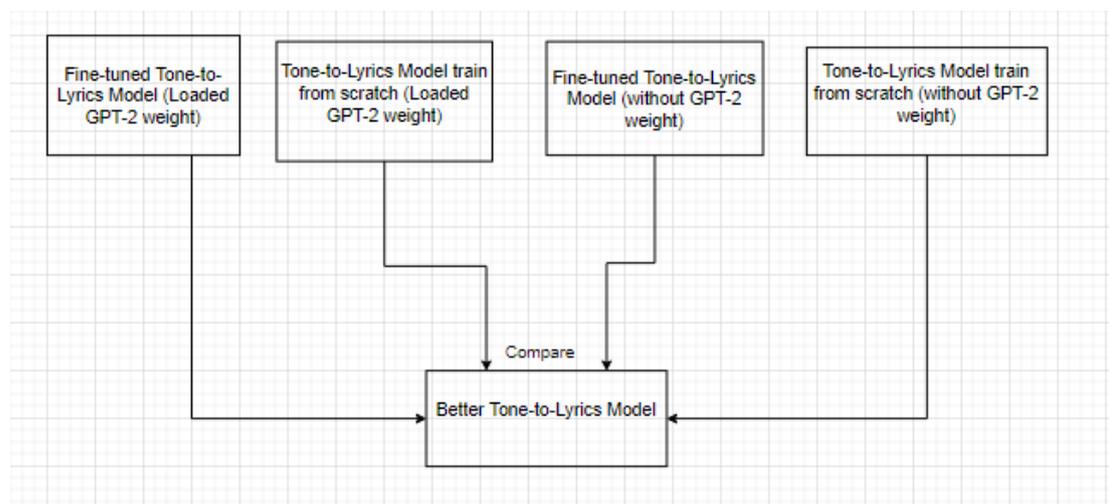


Figure 4.2 Flow chart of Bart models comparison

The second stage focuses on building a Cantopop-specify model which takes tones as input and fills meaningful lyrics that match the tones as output. Making use of Transformer-based encoder decoder model to build a seq2seq model,

4.6 Model Evaluation Metrics

4.6.1 BLEU

BLEU, Bilingual Evaluation Understudy, is one of the metrics that able to evaluate the performance of model on NLP tasks. It can be applied into various types of tasks including translation, text generation, etc. It will give a score comparing the different between the candidate text and the reference text [1]. A higher score indicates better performance of the model.

BLEU is usually applied to evaluate the performance of machine translation. Some people may question about the ability for it to evaluate the performance of text generation as it only cares about the difference between reference text and candidate text. When we talk about text generation, we want the model to generate some new things instead of generating exactly the same as the reference texts. Also, it lacks the ability to consider about different aspects like the emotions, fluency, etc.

However, as it is still a very common metric applying on many NLP tasks, we can still take it as a reference. Therefore, in this project, BLEU will still be used as one of the metrics to evaluate the performance of the model but instead of using it to evaluate the quality of generated text, we can use it as the reference to see how much the model learns from the data or even detect the overfitting problem of the model.

4.6.2 Perplexity

Perplexity is one of the metrics to evaluate the performance of language models. The definition of perplexity of a language model is, inverse probability of the test set, normalized by the number of words. Below equations Eq. (1) and Eq. (2) explaining about the perplexity [22].

$$\begin{aligned} PP(W) &= P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}} \end{aligned} \quad (1)$$

Applying chain rule into above equations to expand the probability of W:

$$PP(W) = \sqrt[N]{\prod_{i=1}^N \frac{1}{P(w_i | w_1 \dots w_{i-1})}} \quad (2)$$

To put words in simple, perplexity is evaluating the performance of the model on predicting the next word. Having a higher probability to predict a word that match the text dataset, meaning the better performance of the model. Notice that, as perplexity is using the inverse probability, so low perplexity indicates better performance of a language models.

4.6.3 Tone Accuracy

Generating lyrics under the limitation of tones is the final goal of this project. The model is built on top of the tone-based lyrics generation approach. Therefore, it's a must to evaluate the model whether it can generate lyrics matching the input tone.

A simple formula will be applied to calculate percentage of lyrics with correct tone, in Eq. (3).

$$Tone\ Accuracy = \frac{\# \text{ of match tones}}{\# \text{ of input tones}} \quad (3)$$

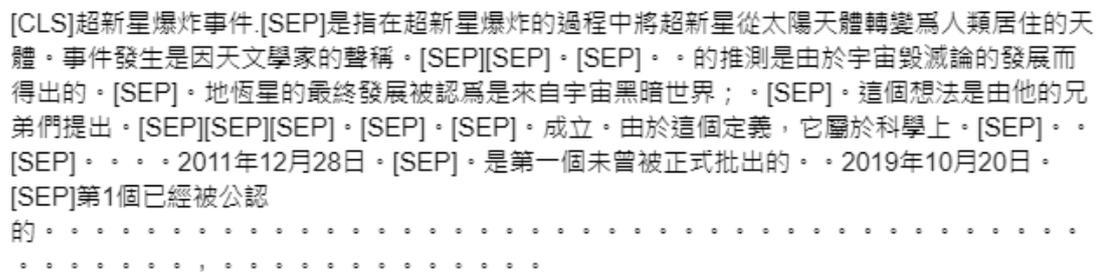
5. Experiments

5.1 Data Crawling

5.1.1 Traditional Chinese Corpus data

Large amount of Traditional Chinese Corpus data is needed for the model pretraining for both GPT-2 model and the Bart model. Therefore, I need to find some high quality Traditional Chinese Corpus data source available online. The corpus data is first crawled from Wikipedia because Wikipedia is considered to contains many high-quality articles [31]. It can provide a large amount of data for the model pretraining. Pretraining the model with the Wikipedia data should give model the ability to generate high-quality text.

However, crawled Wikipedia data is extremely messy containing many abnormal symbols and mixed Traditional Chinese and Simplified Chinese which harm the model, proper cleaning is a must to obtain a high-quality corpus data for model training which requires a lot of effort to do it given that the size of the Wikipedia data is large, and the format of data is not consistent.



[CLS]超新星爆炸事件.[SEP]是指在超新星爆炸的過程中將超新星從太陽天體轉變為人類居住的天體。事件發生是因天文學家的聲稱。[SEP][SEP]。[SEP]。的推測是由於宇宙毀滅論的發展而得出的。[SEP]。地恆星的最終發展被認為是來自宇宙黑暗世界；。[SEP]。這個想法是由他的兄弟們提出。[SEP][SEP][SEP]。[SEP]。[SEP]。成立。由於這個定義，它屬於科學上。[SEP]。[SEP]。2011年12月28日。[SEP]。是第一個未曾被正式批出的。2019年10月20日。[SEP]第1個已經被公認的。 ,

Figure 5.1 Sample output of model learning noise from the data

Above figure shows a sample generated text by the GPT2 model trained with the Wikipedia data. We can see that the sample text contains many abnormal tokens and punctuations which indicates that the model learns the noise from the dataset show it can't generate high-quality text.

Therefore, I finally gave up the Wikipedia data and try to search for other corpus data source. The final corpus data that is used to do the model pretraining is the CC-100 Traditional Chinese dataset, which is a cleaned Traditional Chinese dataset

available on the Internet which is from Common Crawl corpus [30]. The CC-100 Traditional Chinese dataset can directly be used to do the model training.

5.1.2 Lyrics Data

Lyrics data is one of the most important parts for this project.

There isn't any well clean Cantopop lyrics data available online. I need to crawl the data from different sources and do the cleaning by myself. In this project, the lyrics data is mainly come from two websites, Genius and Mojim [32][33].

Compared to the Traditional Chinese corpus data, the number of lyrics data is much smaller and hence, the lyrics data must be much cleaner, no non-sense text and symbol should be allowed in the lyrics data in order to avoid the model learning from the noise easily.

Finally, total 15428 song lyrics are crawled.

5.2 Data Preprocessing

(Notes: as Traditional Chinese corpus data is too large to showcase, lyrics data would be taken as example below for data preprocessing)

5.2.1 Data Cleaning

Data Cleaning is always the most difficult part to do in a machine learning project. As those data crawling from the Internet is usually user-generated content and hence, it is unformatted and extremely messy. As the information in the data is where the model will directly learn from, if there are many noises presenting in the data, such noises will also be learnt by the model and hence, the generated result by the model will often contain various noise as well.

你好嗎)

你 想 去 哪 裡 有 著 我 該 去 哪 裡
想 去 哪 裡 有 著 你 快 樂 探 索
可 否 陪 著 我 讓 自 己 沉 下 去

很 多 人 談 戀 愛 太 遲
每 個 人 都 被 寂 寞 吞 食 了 自 憐 甚 麼 的
只 是 你 身 在 發 呆
誰 把 我 的 情 感 你 想 起 來 誰 忘 記
人 生 很 短 很 美 我 怎 麼 忘 記

Figure 5.2 Sample lyrics generated from a non-well-trained model

Above figure shows one example of generated lyrics from a non-well-trained model. You can see that an abnormal close parenthesis is generated as the output. This is because the data that fit to training stage contains such noises. The model learns the pattern and reproduce it as the output.

The above example showcases the importance of data cleaning to obtain a formatted, organized dataset for model training. To increase the performance of the model, preparing a clean dataset is always the first step as well as the most important step to do so. Both traditional corpus data and lyrics data contains such noises that needed to be cleaned in order to fit it to training stage.

Here is an example of lyrics cleaning:

Before cleaning:

陳奕迅

十年

作詞：林夕
作曲：陳小霞
編曲：陳維陽

如果那兩個字沒有顫抖
我不會發現 我難受
怎麼說出口 也不過是分手

如果對於明天沒有要求
牽牽手就像旅遊
成千上萬個門口
總有一個人要先走

* 擁抱既然不能逗留
何不在離開的時候
一邊享受 一邊淚流

十年之前 我不認識你
更多更詳盡歌詞 在 ※ Mojim.com 魔鏡歌詞網
你不屬於我 我們還是一樣
陪在一個陌生人左右
走過漸漸熟悉的街頭

十年之後 我們是朋友
還可以問候 只是那種溫柔
再也找不到擁抱的理由
情人最後難免淪為朋友

REPEAT * #

直到和你做了多年朋友 才明白我的眼淚
不是為你而流 也為別人而流

[00:02.18]陳奕迅 - 十年
[00:14.79]如果那兩個字沒有顫抖
[00:19.02]我不會發現 我難受
[00:22.28]怎麼說出口 也不過是分手
[00:29.97]如果對於明天沒有要求
[00:34.87]牽牽手就像旅遊
[00:37.85]成千上萬個門口
[00:41.50]總有一個人要先走
[01:57.00][00:47.56]擁抱既然不能逗留
[02:00.51][00:50.83]何不在離開的時候
[02:03.54][00:53.75]一邊享受 一邊淚流
[02:10.79][01:00.70]十年之前 我不認識你
[02:14.53][01:04.73]你不屬於我 我們還是一樣
[02:19.03][01:09.26]陪在一個陌生人左右
[02:22.66][01:12.92]走過漸漸熟悉的街頭
[02:26.26][02:11.02]十年之後 我們是朋友
[02:29.89][01:20.34]還可以問候 只是那種溫柔
[02:34.44][01:24.64]再也找不到擁抱的理由
[02:38.30][01:28.66]情人最後難免淪為朋友
[02:48.70]直到和你做了多年朋友 才明白我的眼淚
[02:55.37]不是為你而流 也為別人而流

Figure 5.3 Sample lyrics to be cleaned crawled from the Internet

The above figure shows a sample lyric crawled from the Internet. We can see how that the data is very messy. It includes many information that we don't want including the name of singer, the song name, related contributors, etc. It even includes some non-sense time code that we don't even know the meaning of it. We don't want the model to learn such non-sense information. Therefore, we need to delete all these out to obtain only the body part of lyrics. However, we can see that even the lyrics body itself contains some extra symbols or even some not related text. Notice that, it's only one example that extracted from the dataset.

After Cleaning:

如果那兩個字沒有顫抖←
我不會發現 我難受←
怎麼說出口 也不過是分手←
←
如果對於明天沒有要求←
牽牽手就像旅遊←
成千上萬個門口←
總有一個人要先走←
←
懷抱既然不能逗留←
何不在離開的時候←
一邊享受 一邊淚流←
←
十年之前 我不認識你←
←
你不屬於我 我們還是一樣←
陪在一個陌生人左右←
走過漸漸熟悉的街頭←
←
十年之後 我們是朋友←
還可以問候 只是那種溫柔←
再也找不到擁抱的理由←
情人最後難免淪為朋友←
←
直到和你做了多年朋友 才明白我的眼淚←
不是為你而流 也為別人而流←
←

Figure 5.4 Sample lyrics to after cleaning

Above figure shows the result after passing it to the cleaner.

The most complicated part of cleaning the lyrics data is that there isn't a consistent format for the lyrics crawled. As all lyrics is upload the by user, there isn't a strict format for the content. Therefore, the cleaner need to take care all these different cases. I need to keep increase the capability of the cleaner cleaning lyrics data in different format in order to obtain the clean lyrics.

One extra thing to notice for the dataset is that, as it's quite common that Cantopop lyrics consists of some other languages such as English, Japanese, or even Korean. As these languages aren't tones languages, and the final model takes only the tone as the input, they needed to be eliminated as well.

Below figure shows the final cleaned lyrics dataset and total 15428 song lyrics are clean and gathered to be the dataset of the project.

:

	lyrics
0	某個他真與假\n信你部分得到\n醉了嗎上我家\n真衣解帶遺下仰慕\n躲於被窩跟我跳舞\n輕撫不需急躁...
1	某個他真與假\n信你部分得到\n醉了嗎上我家\n真衣解帶遺下仰慕\n躲於被窩跟我跳舞\n輕撫不需急躁...
2	世界大亂要撥開風沙找到你到絕處去躲\n意志薄弱要決心紛擾之中找到你到老與天荒\n接近你也不...
3	愛你最好拋開你\n腳印如路過雲地\n事後記不起純屬片刻之美\n永遠帶不走的你\n朦朧情動有趣...
4	除非將畢生所信也推翻\n除非能說服我只關一隻眼\n雖則你和我已是爭吵慣\n常和好於當晚這次諒...
...	...
15423	男誰人能料愛會這樣盼你會體諒\n從前承諾已變了樣愛意那可強\n默默望著滿面淚痕仍然無怨\n怎...
15424	男別逃避了孩子正嚶泣看誰人會關心靠近\n女問候和愛將天意變改不要緊請跟我同行\n男無論太遠還...
15425	沒有月亮我們可以看星光\n失去星光還有溫暖的眼光\n抱著希望等待就少點感傷\n彷彿不覺得寒夜...
15426	若你經已很餓搵到位安樂坐好\n宣佈好事起菜你已等到\n就快嚙嚙你要點醋要咁先似宴會\n滾得高...
15427	電影花木蘭粵語版\n像我嗎若我上了妝是否更漂亮\n我卻太不擅長\n就算想令各位親友拍掌\n但...

15428 rows × 1 columns

Figure 5.5 Final lyrics dataset

5.2.2 Tone2Text dataset building

In the second stage of this project, tone2text model is trained and hence, tone2text datasets need to be prepared. Actually, no extra sources are needed to prepare the tone data. The tone data can be directly extracted from the text data itself using a python library called pycantonese [34].

As mentioned in Methodology, the tone data would be extracted from the text data, and they will be merged to form the tone2text dataset.

Original Text:

Figure 5.6 Sample lyrics to be extracted tone from it

Corresponding tone:

Figure 5.7 Sample tone to extracted tone from lyrics

Figure 5.6 and Figure 5.7 show an example of extracting the tones from lyrics. Figure 5.6 is the lyrics of the song and Figure 5.7 is the extracted tone associated to the original lyrics. Each number is representing the corresponding tone of the

Chinese character at the same position from original text. Each tone number is spaced for proper tokenization. All other characters like space and new line character are preserved.

Tone2text dataset:

```
{
  "tone": "4 2 1 1 1 2 1 1 5 1 2 \n 4 2 1 1 1 2 1 1 5 2 5 \n 4 2 1 1 4 6 2 1 5 1 6 \n 6 1 2 3 6 5 6 6 6 2 2 1 3
1 2 \n \n 4 2 1 2 5 2 2 1 5 1 3 \n 4 2 1 1 6 2 1 1 5 1 2 \n 4 2 1 1 4 6 3 4 2 1 1 4 6 5 1 6 \n 5 2 5 4 6 2 6 3 2
\n \n 4 4 5 5 6 4 3 2 \n 5 3 4 5 1 6 5 6 4 \n 4 4 3 5 4 4 3 2 \n 2 1 4 4 4 3 5 3 6 4 6 \n \n 4 4 6 3 1 1 2 2 1
4 4 4 5 3 3 3 5 1 1 \n 6 5 1 1 4 4 6 1 1 2 2 1 1 2 2 1 5 3 4 4 3 \n 4 4 6 5 1 2 1 2 1 4 4 4 3 5 3 3 5 1 1 \n 3 5 1
2 4 4 6 2 2 2 3 1 2 1 3 2 3 \n \n \n 1 1 1 2 3 2 1 1 3 2 3 1 1 1 1 2 1 2 2 5 1 2 \n 4 6 1 1 4 6 5 4 2 1 1 4 6 3
1 6 \n 5 2 3 4 6 5 6 5 2 \n \n 6 4 2 1 4 6 2 5 \n 2 3 4 4 2 6 5 3 1 \n 4 4 1 2 5 4 1 6 \n 5 5 6 4 2 1 2 2 5 3
2 \n \n 4 4 6 3 1 1 2 2 1 4 4 4 5 3 3 3 5 1 1 \n 6 5 1 1 4 4 6 1 1 2 2 1 1 2 2 1 5 3 4 4 3 \n 4 4 6 5 1 2 1 2 1 4 4
4 3 5 3 3 5 1 1 \n 3 5 1 2 4 4 6 \n 2 1 1 5 3 4 3 3 2 1 1 3 3 4 5 3 \n 2 1 2 3 5 4 6 3 5 6 \n \n 4 4 6 3 1 1 2 2 1
4 4 4 5 3 3 3 5 1 1 \n 3 3 1 2 4 5 6 1 2 2 3 5 4 6 3 1 5 5 4 5 3 \n 4 4 6 5 1 2 1 2 1 4 4 4 3 5 3 3 5 1 1 \n 5 3 1
2 4 5 3 6 5 5 1 2 3 6 3 6 4 1",
  "lyrics": "如果爭執 偏口交給我修好\n如果翻風 關起窗給你擁抱\n如果不安 誠實請給我知照\n做不好再做 無奈並未代表所需要的好\n\n如
果紛擾 耳朵只聽你傾訴\n如果偏風 病菌分給我都好\n如果焦急和暴躁 如果消失能令你息怒\n我可以迴避到避世島\n\n旁人 眼裡 為人 再好\n
與愛情無關沒有用途\n投懷 送抱 然而 跌倒\n好的傻瓜忘記了愛是殘酷\n\n原來奉獻多多少少依然徒勞兩個世界也分開\n\n任我花精神期待 清
清楚楚辛辛苦苦不會帶來憐愛\n原來是我多此一舉傷痕累累要我尷尬也應該\n\n對你的好仍存在 好好相處不等於愛 誰愛\n\n\n失足深海假使得一個
水泡 風波之中首先只想你安好\n拿驕犧牲維護你 情感之中純屬寄生族\n\n你只要存在我便會好\n\n為何 苦心 成為 苦惱\n這愛情城堡未免太高
\n如何 得到 無從 知道\n你腦內浮現的總比我更好\n\n原來奉獻多多少少依然徒勞兩個世界也分開\n\n任我花精神期待 清
清楚楚辛辛苦苦不會帶來憐愛\n原來是我多此一舉傷痕累累要我尷尬也應該\n\n對你的好仍存在\n好先生與愛情競賽 好先生太過難被愛\n始終相信你存在故我在\n\n原來
奉獻多多少少依然徒勞兩個世界也分開\n最固執感情無奈 一起相處有權任性不會有權被愛\n原來是我多此一舉傷痕累累要我尷尬也應該\n\n你要
的好難被替代 我有的好最後化做 塵埃"
```

Figure 5.8 Sample data for tone2text dataset

A lyric and the corresponding tone are combined to be treated as one data. All such combinations are stored in json to be the dataset. Figure 5.8 show an instance of the tone2text dataset.

5.2.3 Tokenization

In usual case, each model has a corresponding tokenizer. However, in this project, we mainly deal with the Chinese character and both original projects of GPT-2 and Bart doesn't support Chinese character tokenization. There are two ways to handle such situation:

1. Train a new tokenizer with the corpus data
2. Use a trained tokenizer from another model

As Bert model from Google has a pretrained Chinese tokenizer and it can cover most of the cases of our data. Bert tokenizer can be directly used to do the tokenization.

5.3 GPT2 Models

As describe above, there would be two different GPT2 models, one is trained from scratch with lyrics data only, another fine-tuned a pretrained model with traditional Chinese corpus data.

5.3.1 Structure

Although adjusting the model can lead to a very different performance of the models, as the comparison approaches are comparing the performance of models trained using pre-training and fine-tuning and training from scratch. Also, due to the limitation of hardware resources, it's basically impossible to train a large model with large amount of data. There, the model structure of model will be relatively a small model and remain the same in this experiment to highlight the difference between these three approaches.

```
GPT2Config {
  "activation_function": "gelu_new",
  "architectures": [
    "GPT2LMHeadModel"
  ],
  "attn_pdrop": 0.1,
  "bos_token_id": 50256,
  "embd_pdrop": 0.1,
  "eos_token_id": 50256,
  "gradient_checkpointing": false,
  "initializer_range": 0.02,
  "layer_norm_epsilon": 1e-05,
  "model_type": "gpt2",
  "n_ctx": 1024,
  "n_embd": 768,
  "n_head": 12,
  "n_inner": null,
  "n_layer": 6,
  "n_positions": 1024,
  "output_past": true,
  "reorder_and_upcast_attn": false,
  "resid_pdrop": 0.1,
  "scale_attn_by_inverse_layer_idx": false,
  "scale_attn_weights": true,
  "summary_activation": null,
  "summary_first_dropout": 0.1,
  "summary_proj_to_labels": true,
  "summary_type": "cls_index",
  "summary_use_proj": true,
  "task_specific_params": {
    "text-generation": {
      "do_sample": true,
      "max_length": 320
    }
  },
  "tokenizer_class": "BertTokenizer",
  "torch_dtype": "float32",
  "transformers_version": "4.12.2",
  "use_cache": true,
  "vocab_size": 21128
}
```

Figure 5.8 Configuration of GPT-2 model

Above figure illustrating the configuration of the GPT-2 model [18].

1. 6 hidden layers
2. 12 attention heads
3. 1024 maximum sequence length
4. 1024 dimensions of casual mask
5. 1024 dimensions of hidden state and embeddings
6. Using Bert tokenizer and having 21128 vocab size

5.3.2 Statistics

As mentioned in Methodology, two different GPT-2 models are trained using training from scratch and pre-training and fine-tuning respectively and this section show the statistics of two models, GPT-2 trained from scratch and GPT-2 fine-tuned from a pretrained model.

Training Statistics:

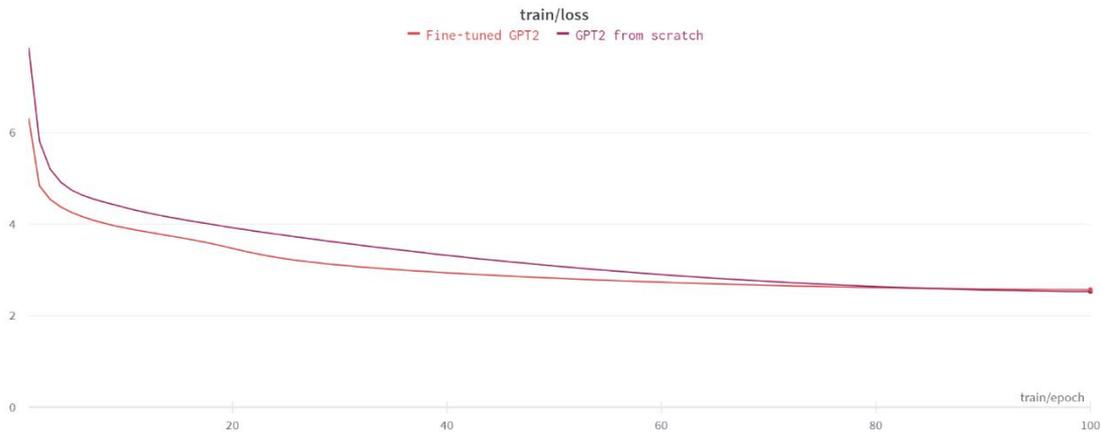


Figure 5.9 Training loss of GPT-2 models

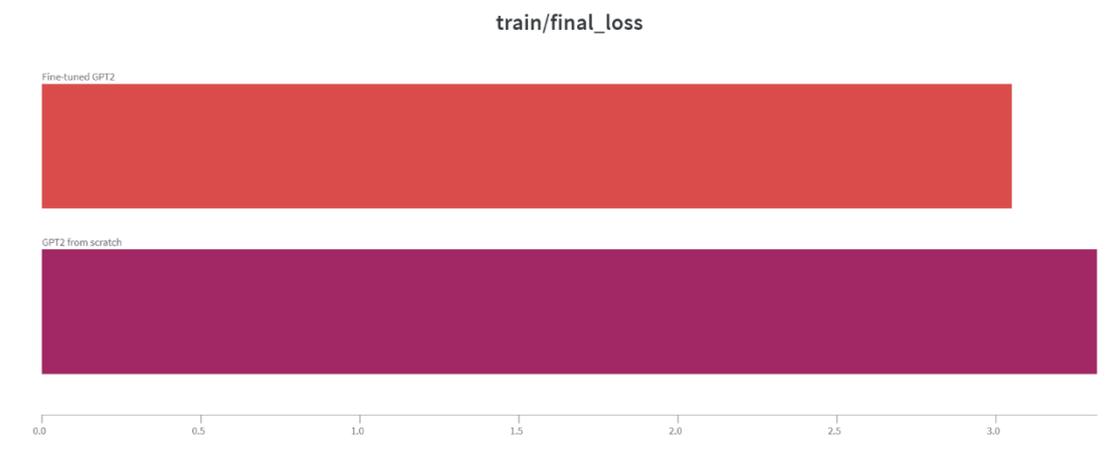


Figure 5.10 Final Training loss of GPT-2 models

Validation Statistics:

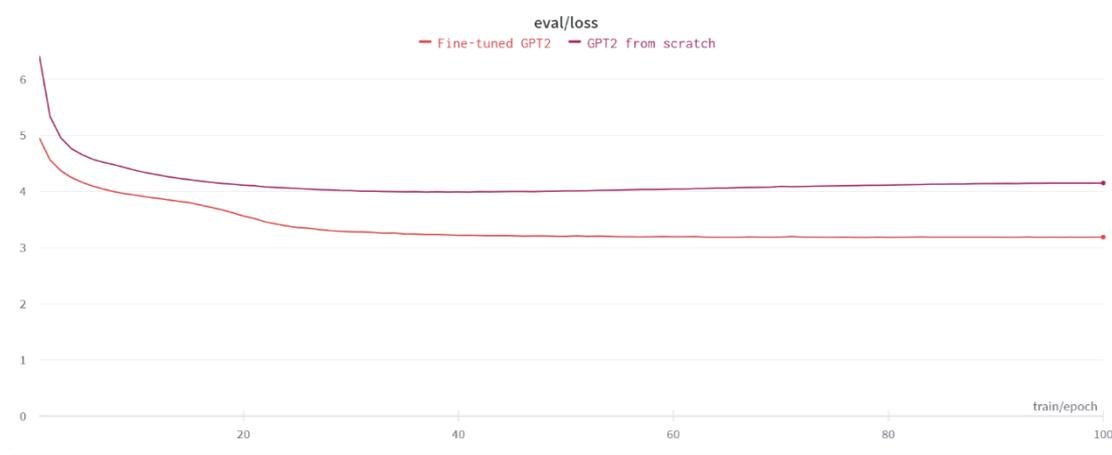


Figure 5.11 Validation loss of GPT-2 models

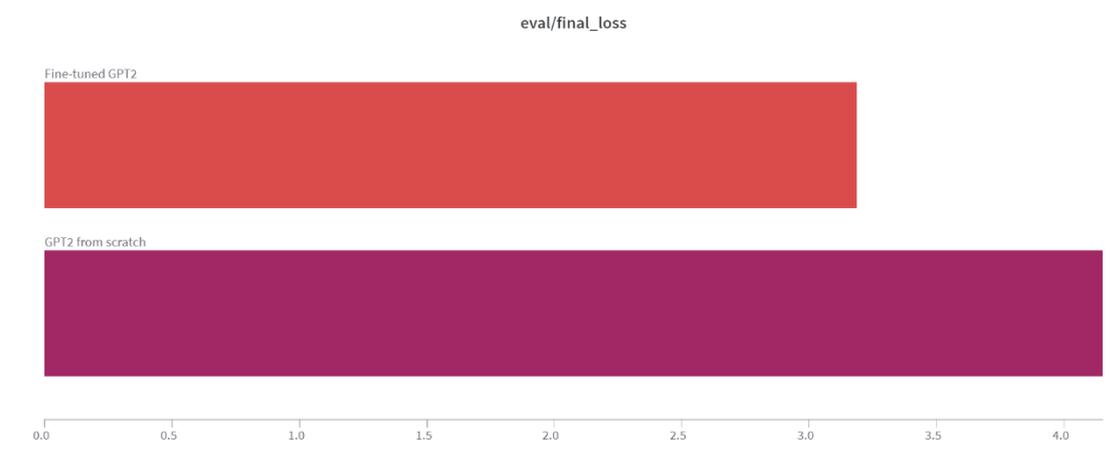


Figure 5.12 Final Validation loss of GPT-2 models

Above figures show the statistics of two models, Figure 5.9 and Figure 5.10 representing the training statistics while Figure 5.11 and Figure 5.12 representing the validation statistics. Generally speaking, we can observe that fine-tuned GPT2 model has a lower initial loss and converges faster during training under the same configurations. The final loss of fine-tuned is lower than scratch GPT2 model in both Training stage and Validation stage. Although loss aren't the performance indicator of the models, it can still show that fine-tuned GPT2 model fit to the data better compared to scratch model. This is due to pretrained-model already learn some representation of Traditional Chinese during pre-training with corpus data.

5.4 GPT-2 Evaluation

As described above, various metrics are applied to evaluate the performance of the models. GPT-2 doesn't take tone as input, so tone accuracy is not applicable to GPT-2 Evaluation. BLEU and Perplexity will be applied, and the results are as follow:

5.4.1 BLEU

	GPT2 trained from scratch	Fine-tuned GPT2
BLEU	0.2	4.8

Above table shows the BLEU scores of two GPT-2 models. We can observe that there is a big difference between the score of GPT-2 trained from scratch and Fine-tuned GPT-2. Previous one got 0.2 score and later one got 4.8 score. An higher BLEU score indicates a better performance and hence, Fine-tuned GPT-2 performs better than GPT2 trained from scratch in term of BLEU.

5.4.2 Perplexity

	GPT2 trained from scratch	Fine-tuned GPT2
Perplexity	343.5803	235.6772

From the above table, we can see that the perplexity score of fine-tuned GPT2 is lower than the score of GPT2 trained from scratch. Based on the definition of perplexity, lower score indicates better performance. Therefore, Fine-tuned GPT2 performance better than GPT2 trained from scratch in term of perplexity.

5.4.3 Sample

Although it's hard to do human evaluation in this project, it's still worth to explore the example output of two models.

Starting Sentence:

你好嗎，為誰大聲哭過

(How are you. Are you crying for someone?)

Generated by GPT-2 trained from scratch

Example 1:

你好嗎為誰大聲哭過
仍舊望見著他日讓我無法再沒法再學會想起
人要我在你的心底是我的夢想起
你曾如果沒有想念掛念過去心裡都市流浪與她不太苦痛楚
寧願放下次放任何時間無法再等待你在嘆息都知道你痛苦等待誰人為止痛楚
心痛哭泣但無遠難道你不甘心為何結束我

Figure 5.13 Example 1 generated by GPT-2 trained from scratch

Example 2:

你好嗎為誰大聲哭過
我無法躲回家的快樂同享受不知道你是否再是我的感受傷害怕受
你和我愛人的回憶今天大聲音那長久我的最尾不必再也不是否期待一瞬間陪在人間陪你過程、
等待一個人孤獨個人笑中尋覓那麼長久我的最尾不必再也不是否期待一瞬間陪在人間陪你過程、

Figure 5.14 Example 2 generated by GPT-2 trained from scratch

Example 3:

你好嗎為誰大聲哭過
那當初當初只可愛想起妳我和妳回頭想起跟妳重拾起
讓我們想念去點起離別了當初吻過了這段奇
曾經離開始給我獨個你離開始至少記憶最初衷心不敢再想起離開始至此刻回憶回頭回憶去年妳的身邊我的
心為何獨處

Figure 5.15 Example 3 generated by GPT-2 trained from scratch

GPT-2. Generally speaking, we can see that the output generated by the fine-tuned GPT-2 model have a more lyrics-like structure. GPT-2 trained from scratch tends to generate long sentence which is difficult to fit into the melody because melody is built block by block and the pace is kept changing with the use of different music notes. Therefore, long sentence can hardly form a block of lyrics to fit in the melody. Fine-tuned GPT-2 tends to generate lyrics that is composed by short sentences which is suitable for fitting it into melody block. Therefore, we can conclude that the quality of lyrics generated by fine-tuned GPT-2 is better than the lyrics generated by GPT-2 trained from scratch.

5.4.4 Comparison

To conclude, fine-tuned GPT-2 shows a better overall performance compared to GPT-2 trained from scratch. Therefore, fine-tuned GPT-2 will be used in the Bart model phase.

5.5 Bart Model

5.5.1 Structure

As mentioned above, the Bart model structure used tends to be a small structure and will be kept as the same for all the models to highlight the difference between pretraining and fine-tuning and training from scratch.

```
BartConfig {
  "_name_or_path": "models/checkpoint-13000",
  "activation_dropout": 0.1,
  "activation_function": "gelu",
  "architectures": [
    "BartForConditionalGeneration"
  ],
  "attention_dropout": 0.1,
  "bos_token_id": 101,
  "classifier_dropout": 0.0,
  "d_model": 768,
  "decoder_attention_heads": 12,
  "decoder_ffn_dim": 3072,
  "decoder_layerdrop": 0.1,
  "decoder_layers": 6,
  "decoder_start_token_id": 102,
  "dropout": 0.1,
  "early_stopping": true,
  "encoder_attention_heads": 12,
  "encoder_ffn_dim": 3072,
  "encoder_layerdrop": 0.1,
  "encoder_layers": 6,
  "eos_token_id": 102,
  "forced_eos_token_id": 102,
  "gradient_checkpointing": false,
  "id2label": {
    "0": "LABEL_0",
    "1": "LABEL_1",
    "2": "LABEL_2"
  },
  "init_std": 0.02,
  "is_encoder_decoder": true,
  "label2id": {
    "LABEL_0": 0,
    "LABEL_1": 1,
    "LABEL_2": 2
  },
  "max_length": 256,
  "max_position_embeddings": 1024,
  "model_type": "bart",
  "num_hidden_layers": 6,
  "pad_token_id": 0,
  "scale_embedding": false,
  "tie_word_embeddings": 0,
  "tokenizer_class": "BertTokenizer",
  "torch_dtype": "float32",
  "transformers_version": "4.12.2",
  "use_cache": true,
  "vocab_size": 21128
}
```

Figure 5.19 Configuration of Bart model

Above figure illustrating the configuration of the Bart model.

The configuration of decoder is the same as the configuration of GPT2 model for properly loading the weight of GPT2 as the initial weight of decoder.

The configuration of encoder is [35]:

1. 6 encoder layers
2. 12 attention heads
3. 256 maximum sequence length
4. 3072 dimensions of the feed-forward layer
5. Using Bert tokenizer and having 21128 vocab size

5.5.2 Statistics

As mentioned in the methodology, there are four models are trained using different training approaches and this section shows the statistics of the four models, scratch Bart, scratch Bart loaded GPT-2 weight, fine-tuned Bart without loading GPT-2 weight, fine-tuned Bart loaded GPT-2 weight.

Training Statistics:

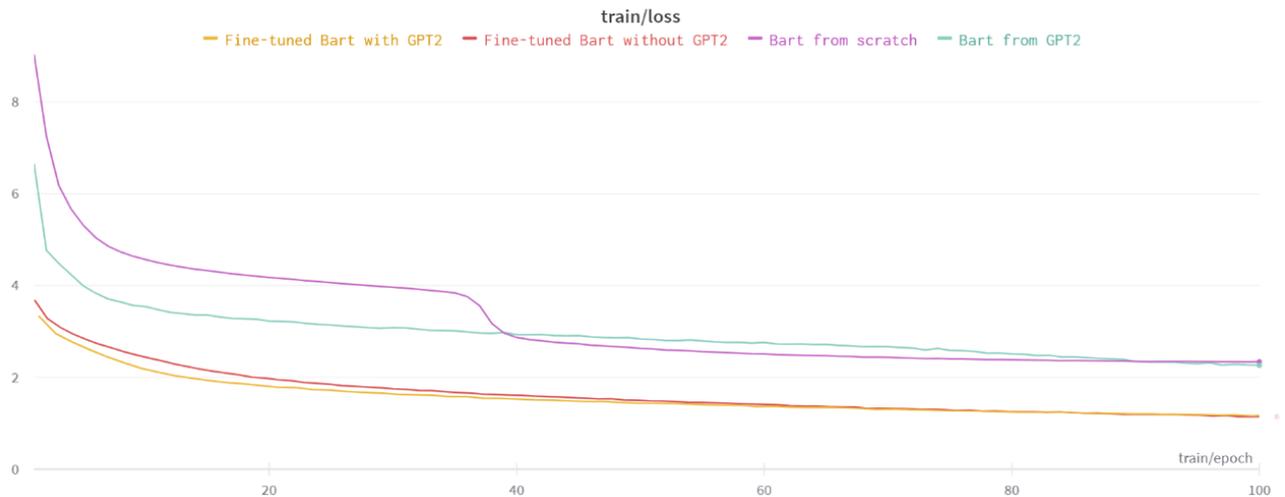


Figure 5.20 Training loss of Bart models

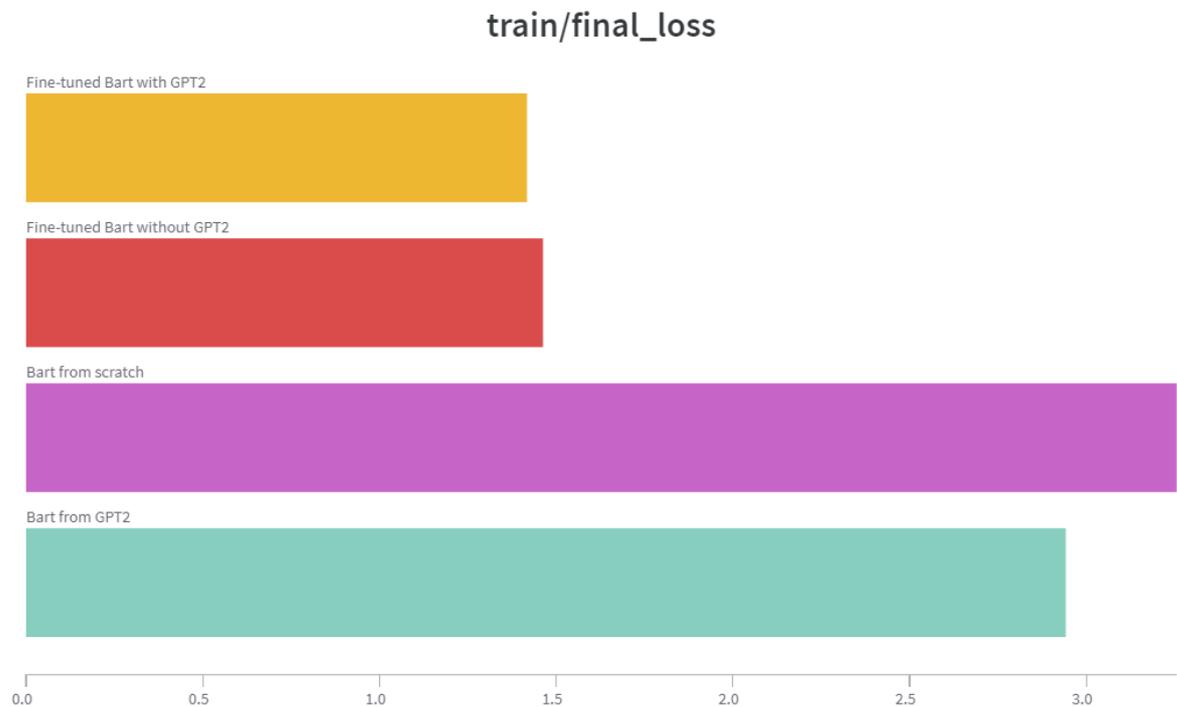


Figure 5.21 Final loss of Bart models

Validation Statistics:

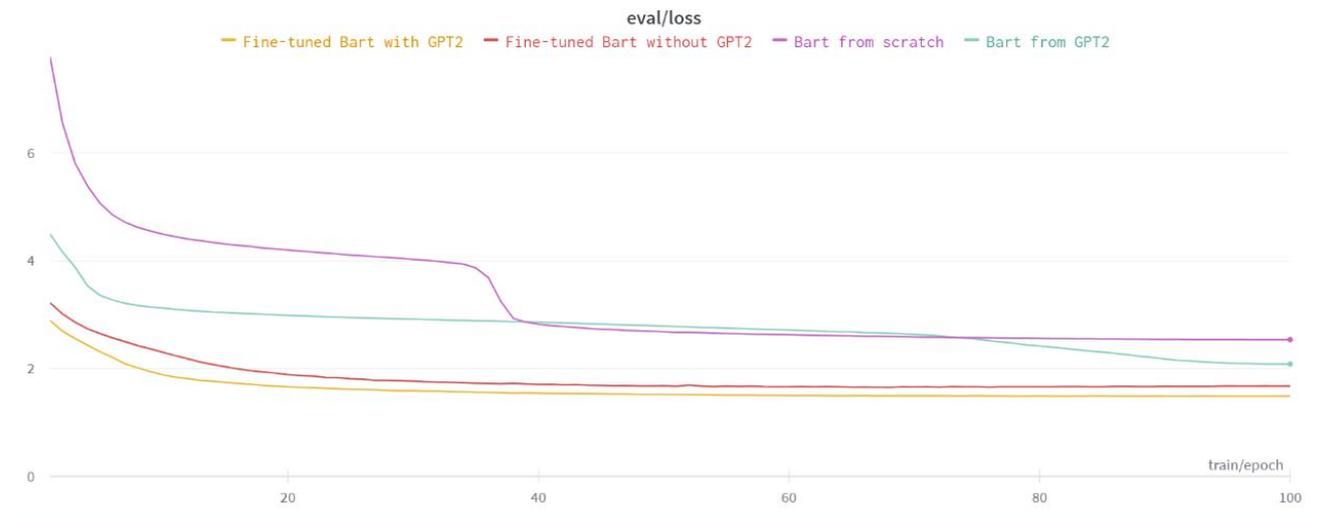


Figure 5.22 Validation loss of Bart models

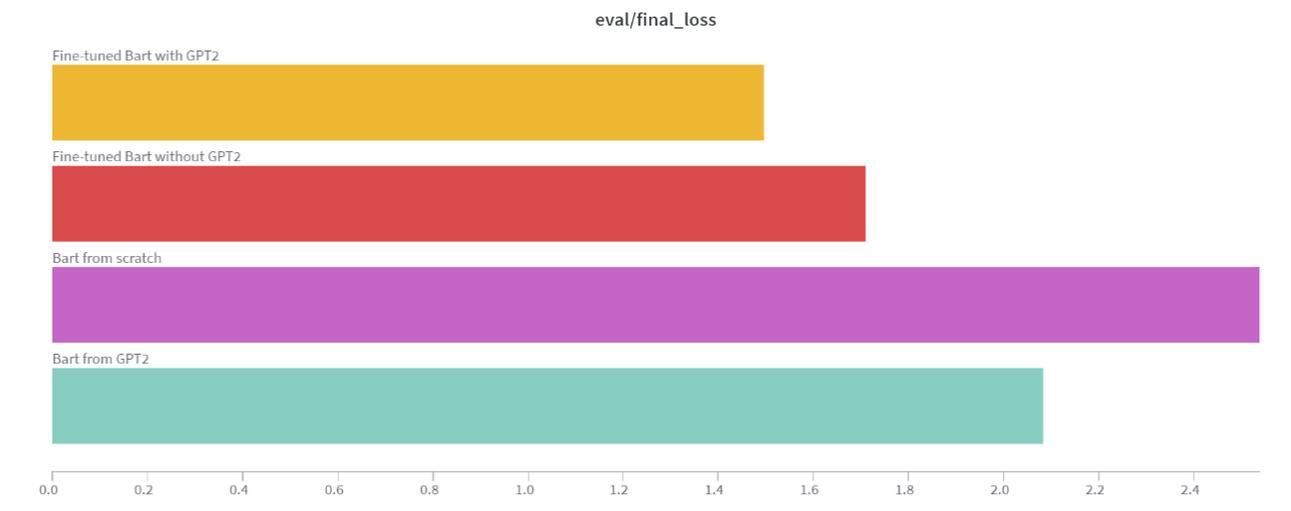


Figure 5.23 Final Validation loss of Bart models

Figure 5.20 and Figure 5.21 is representing the training statistics while Figure 5.22 and Figure 5.23 is representing the validation statistics. We are able to observe from the statistics that fine-tuning the model from a pretrained model has a big advantage during training. Fine-tuned models outperform the models trained from scratch no matter in terms of initial loss, convergence rate or final loss. Also, we can observe that loading the weight of pre-trained GPT-2 model give a little improvement as well.

5.6 Bart Evaluation

5.6.1 BLEU

	Bart trained from scratch	Bart trained from a model loaded GPT-2 weight	Bart fine-tuned from a pretrained model	Bart fine-tuned from a pretrained model loaded GPT-2 weight
BLEU	0.3	0.3	0.7	0.8

Above table shows the BLEU scores of four Bart models. We can see that fine-tuned Bart models give a higher BLEU score compared to Bart models trained from scratch. Previous one got 0.7 and 0.8 score and later one got 0.3 score. A higher BLEU score indicates a better performance and hence, Fine-tuned Bart models performs better than Bart models trained from scratch in term of BLEU. However, Loading the weight of GPT-2 doesn't give a significant improvement in term of BLEU score. Bart fine-tuned from a pretrained model loaded GPT-2 weight gives a improvement of 0.1 score. Bart trained from a model loaded GPT-2 weight doesn't even have any improvement compared to the model trained from scratch without GPT-2 weight.

5.6.2 Perplexity

	Bart trained from scratch	Bart trained from a model loaded GPT-2 weight	Bart fine-tuned from a pretrained model	Bart fine-tuned from a pretrained model loaded GPT-2 weight
Perplexity	674.8038	635.6714	344.2669	337.1880

Above table shows the perplexity scores of each model. We can see that Bart fine-tuned from a pretrained model which loaded GPT-2 weight get the lowest score and Bart trained from scratch get the highest score which means the previous one performs the best under this metrics. We can observe that there is a big difference in the score between models fine-tuned from a pretrained model and models without pre-training. Also, there is also a small improvement when we load the weight of pre-trained GPT-2 model into the Bart model. This indicates the power of pre-training and fine-tuning.

5.6.3 Tone Accuracy

	Bart trained from scratch	Bart trained from a model loaded GPT-2 weight	Bart fine-tuned from a pretrained model	Bart fine-tuned from a pretrained model loaded GPT-2 weight
Tone Accuracy	0.98975	0.9877	0.99655	0.9938

Above table shows the tone accuracy of each model. We can see that all 4 models give a very high tone accuracy. All of them are higher than 98% and that maximum difference is less than 1%. Therefore, we can conclude that as long as we train the model with enough steps, the model can obtain a high tone accuracy.

5.6.4 Sample

This section showcases some output samples generated by the above four models. The model takes tones as the input and generate lyrics which should match the tones as output. Two samples of each model are shown in this section to do the comparison. (More samples can be found in Appendix)

Input:

5 1 4 3 1 5 2

4 4 6 6 3 3 1 2 3 5 6 3

6 5 1 2 5 4

4 2 5 6 2 1 3 6

Generated by scratch Bart model without GPT2 weight

Sample 1:

我身從變的吻現
唯求命運氣轉一顆愛你沒變
願你一起你能
而戀你亦始終接近

Sample 2:

無聲其實不會相
其情是別要怪不可算無事過
恨與深相距離
情可會做可不變動

Generated by scratch Bart model loaded GPT2 weight

Sample 1:

我的情節分秒到
情形就像要跳得到跳上就快
為了擔險了迷
眉苦了就想不要亂

Sample 2:

我的頭看不下這
情人沒問過去多久卻也沒怨
像我不想已從
還好你沒怎麼過沒

Generated by fine-tuned Bart model without GPT2 weight

Sample 1:

冷風流過的雨水
原來就是帶著一起向我蕩去
夜裡奔走旅程
回到你在這一個字

Sample 2:

我的微笑不渺小
原來命運要看多少次你重看
沒有知己哪尋
而這已是誰的寄望

Generated by fine-tuned Bart model loaded GPT2 weight

Sample 1:

曙光融化心裡透
遙遙望外雪卻不懂放下內疚
在你的口裡留
留到你在此刻照舊

Sample 2:

無辜來世將永久
流離樹絕世欠生死去了又見
沒有花果也甜
回首也沒感恩怨恨

We can observe that the samples generated by the fine-tuned Bart models give a better quality. Meaningless words or sentences are highlighted in red. Samples generated by scratch Bart model tends to generate lyrics without much sense and it can hardly generate a complete meaningful sentence in each line. Also, it's hardly to observe the relation between each line generated. The samples generated by fine-tuned Bart tends to generate meaningful word and sentences while the relation between each line can be observed and hence, the whole paragraph of lyrics maintain a meaningful content.

5.6.5 Models Comparison

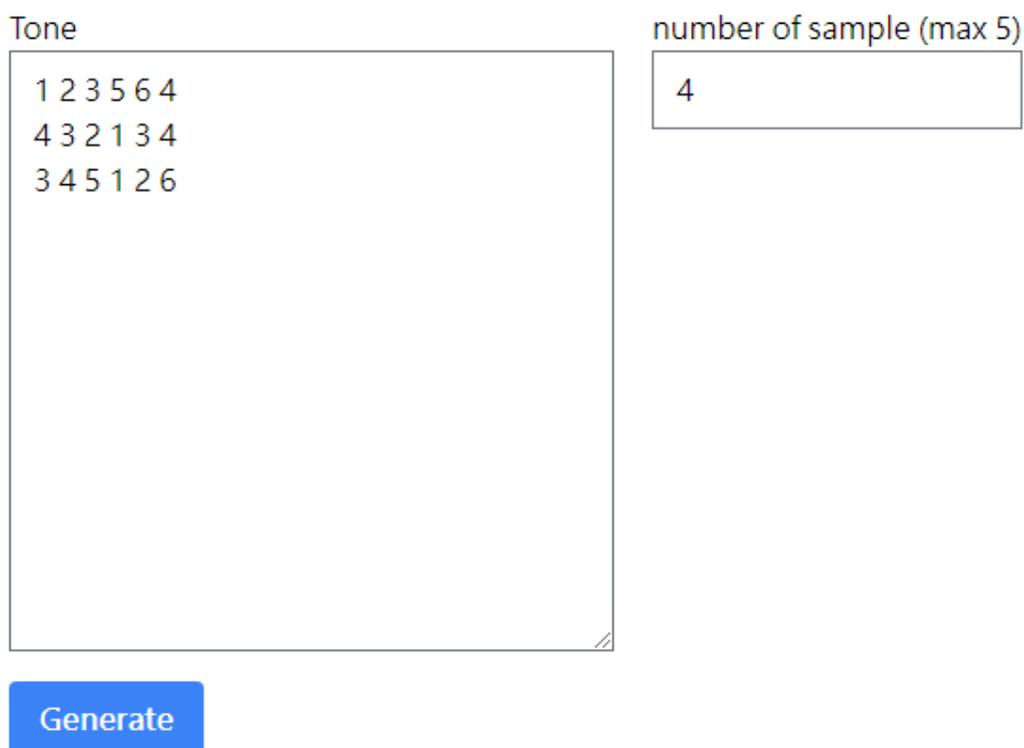
From the above sections, we can see that fine-tuning a model using a pre-trained model outperform model trained from scratch while loading the weight of GPT-2 gives a small improvement which may not be able to be observed significantly but still give some sort of advantage to the model. Pre-trained Bart model already well-learn the representations of tone-to-text relations as well as the contextual information and hence, we can directly fine-tune it for the lyrics generation task with tone2lyrics dataset. Pre-trained GPT-2 learnt about the contextual information of lyrics while it doesn't know about the tone-to-text relations and hence, loading GPT-2 weight can only give a small improvement. To conclude, fine-tuned Bart model which loaded GPT-2 weight gives the best performance in general.

6. Tone2Cantopop

As the final goal of this project is to assist Cantopop Lyric Composition which help public write lyrics. Therefore, an easy-to-use interface for public to access the model with extra helping features, I have created a web tool called Tone2Cantopop for public to use.

6.1 Lyrics Generation

The tool mainly provides the function which allow user to input tone converted from the melody and output lyrics that match the input tone.



The screenshot shows a web interface for lyrics generation. On the left, under the heading "Tone", there is a text area containing three lines of numbers: "1 2 3 5 6 4", "4 3 2 1 3 4", and "3 4 5 1 2 6". To the right, under the heading "number of sample (max 5)", there is a text input field containing the number "4". Below these inputs is a blue button labeled "Generate".

Figure 6.1 Input interface for lyrics generation

Above figure shows the interface for user inputting the tone to load the model and generate the lyrics. Except from inputting tone, there is one extra parameter that user can input which is the number of samples. User can decide how many samples to be generated from the model. Maximum number available currently is 5.



Figure 6.2 Tabs to toggle between generated samples

User can click on the tab number to toggle between the samples generated by the model.

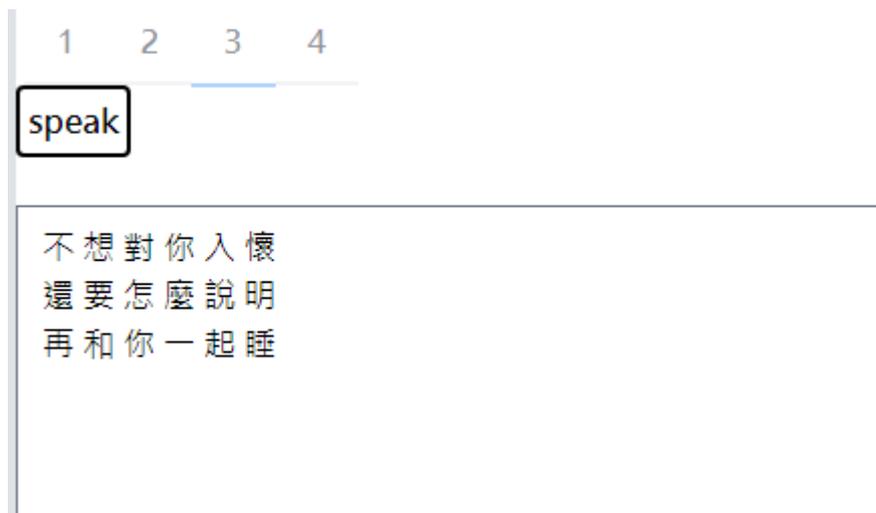


Figure 6.3 Third sample generated

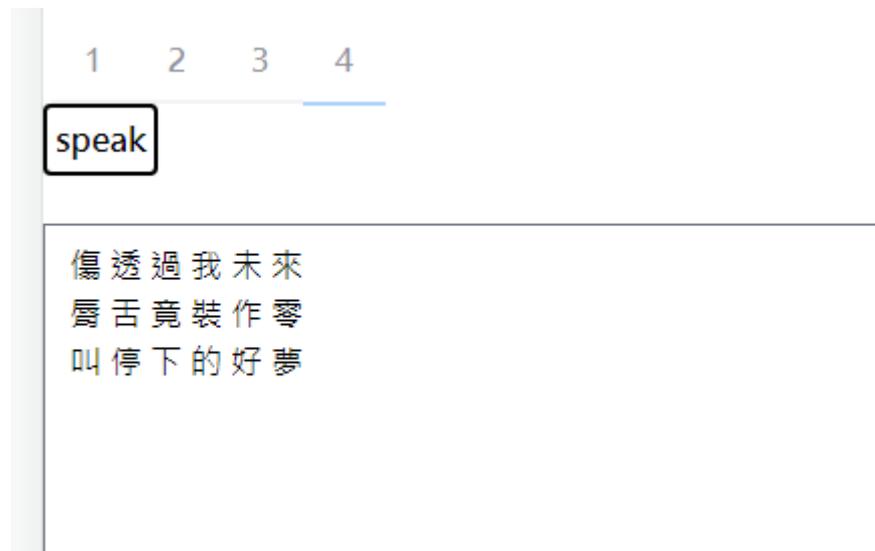


Figure 6.4 Fourth sample generated

Figure 6.3 and Figure 6.4 show the samples toggled between each other. Figure 6.3 refers to the lyrics in tab 3 and Figure 6.4 refers to the lyrics in tab 4.

6.2 Text-to-speech

Text-to-speech function is provided to user for demo purpose. Although providing text-to-sing is the best demonstration, it's another machine learning topic that requires a lot of effort to achieve. Therefore, for current stage, text-to-speech is provided as a simple demo for user check whether the generated lyrics sounds good or not.

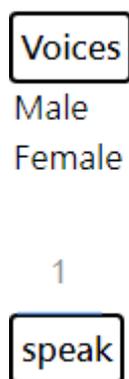


Figure 6.5 Text-to-speech interface

Figure 6.5 refers to the text-to-speech interface. User can choose a male voice or a female voice to do the demonstration and click on the speak button to start the speech. Also, when the word is being spoken, the spoken word will be highlighted for indicating purpose.

我不能夠跟你走
從來沒話說對不起對你做錯
沒有一種理由
如果你是否不接受
我不能已不記得

Figure 6.6 Spoken word highlight feature

Figure 6.6 shows an example of spoken word highlight feature. When the word “走” is being spoken, it's highlighted automatically.

6.3 Tone Comparison

As the goal of this project is inputting tone to model and generating lyrics matching the tone as output, the tool has provided the comparison between the input tone and tone of generated lyrics. Unmatched tone is highlighted, and the total accuracy is calculated as the reference.

Input Tone	Output Tone	Tone Accuracy
1 2 3 5 6 4	1 2 3 5 6 4	88.89%
4 3 2 1 3 4	4 3 2 1 3 4	
3 4 5 1 2 6	3 4 5 2 1 6	

Figure 6.7 Tone comparison feature

Above figure shows a sample output of the tone comparison feature. The unmatched tones 2 1 is highlighted in red and the corresponding accuracy is $16/18 = 88.89\%$. User can then further make modification on the lyrics to make it match the input tone.

7. Limitation

There are some limitations which may affect the results of the models trained.

7.1 Size of dataset

It's hard to tell whether the dataset it's enough for the model. However, it's never wrong that bigger dataset can let the model learn more and better. In this project, about 15000 lyrics data and the same size of tone2lyrics data, 10GB corpus data and 200MB tone2corpus data which is extracted from the corpus are used.

For the lyrics data, although I have mentioned above that the dataset can be expanded to include even Mandarin pop lyrics due to tone-based lyrics generation approach, this project's focus is still Cantopop lyrics and hence, the lyrics data is still built mainly using Cantopop lyrics in this semester. However, it's a possible direction to include Mandarin pop lyrics in the dataset in next semester to compare the results.

For the tone2corpus data, although the origin corpus data is 10GB, only 200MB tone2lyrics data is built because there is computational limitation for extracting the tone and forming the dataset. Therefore, only 200MB data can be prepared in this semester but it's possible to prepare larger tone2corpus data to do the pre-training to see whether the result can be improved.

7.2 Multiple possible tones of a Chinese character

There are some edge cases that when a character is combined with different characters to form a word, the tone of the character will be different.

音節 (香港語言學學會)	粵音	根據	同音字	相關音節	
zung1	[46]	黃(p.47) 周(p.1) 李(p.44) 何(p.342)	鐘, 稷, 癩 [46.]	--選擇--	中土, 中心, 中天[32.]
zung3	[46]	黃(p.47) 周(p.1) 李(p.44) 何(p.343)	鍾, 鍾, 癩 [11.]	--選擇--	中邪, 中尙, 中風[7.]

Figure 7.1 Example of a Chinese character with multiple tones

Source: [36]

Figure 7.1 shows one example that when a Chinese character, 中 (middle) from Figure 7.1, is combined to form different words, 中心 (center) and 中風 (stroke) from Figure 7.1, the tone associated would be different. Tone is 1 with the case 中心 and tone is 3 with the case 中風. In this semester, as we are using Bert tokenizer which handle Chinese in a character-based approach and hence the extraction of tone is also character-based. Therefore, the cases of multiple possible tones of a Chinese character are not taken into consideration. To improve it, we can train our own tokenizer using WordPiece / SentencePiece approach which tokenize the text into word level / sentence level. We can then do the tone extraction based on the word / sentence which should be able to solve the problem of multiple possible tones of a Chinese character.

8. Conclusion

In this project, the power of Transformer architecture is utilized to develop a model specifically fit for generating Cantopop lyrics. Traditional word/sentence-prediction-based lyrics generation and melody-based lyrics generation as they are not suitable for the nature of Cantopop. A brand-new tone-based approach is presented here for Cantopop lyrics generation.

Several models (GPT-2 and Bart) are trained using two different training approaches, pre-training and fine-tuning and training from scratch. We can see the power of having a pretrained model trained in-domain and fine-tuning it for specific task. No matter the training statistics, evaluation metrics score or quality of generated lyrics, fine-tuning the model from a pretrained model gives a much better result comparing to training a model from scratch.

Also, a web application Tone2Cantopop is developed for public to use which provide a several features except from generating lyrics based on the input tone like text-to-speech and input and output tone comparison.

Cantopop is not a popular topic in the field of machine learning due to its universality. This project aims at building a foundation on applying machine learning techniques into Cantopop. Cantopop or Cantonese is interesting topic to work on due to its own unique nature and hopefully this project can raise the interest or attention on Cantopop or Cantonese.

9. Possible Future Development

9.1 Model improvement

9.1.1 Theme/Title based lyrics generation

At current point, the model only accept tone as input and fill in the lyrics randomly. We can't expect what direction or theme that the lyrics is going to be generated. This feature can be implemented by adding a label of the title or theme in front of the data in order to let the model learn to generate lyrics based on the input theme and title. In such way, it provides a bigger flexibility for user to generate lyrics of different themes.

9.1.2 Lyrics generation with partly finished lyrics

Consider the actual case when lyricists writing lyrics, instead of having empty idea, it's more common that they stuck at the middle way which means that they partly finished some lyrics but run out of idea afterward. Therefore, to make the model more applicable to a practical situation. We can improve the model to accept partly finished lyrics with unfilled tones as input and the model should be able to fill in the lyrics that fit the input tones as well as semantically coherent to the finished lyrics.

9.1.3 Model Configuration Adjustment

As the main focus in this semester is comparing the model difference using different training approach, i.e., pre-training and fine-tuning and training from scratch, so the model structure remains the same. In next semester, based on the best training approach, we can adjust the model configuration in order to find the best one to keep improving the quality of generated lyrics.

9.2 App Improvement

9.2.1 Account system

An account system can be implemented that each user can register an account to use the application. Some simple features can be provided to each account. For example, user can save the lyrics generated and review or export the saved lyrics at any time.

9.2.2 Rating system

As the lyrics is auto generated by the model, it's difficult to ensure the quality of generated lyrics, a rating system can be implemented which let user rate the lyrics which can be considered a type of human evaluation on the output results and the rating can be saved for future improvement of the model.

10. Reference

- [1] J. Brownlee, ‘A Gentle Introduction to Calculating the BLEU Score for Text in Python’, *Machine Learning Mastery*, Nov. 19, 2017.
<https://machinelearningmastery.com/calculate-bleu-score-for-text-python/>
(accessed Nov. 13, 2021).
- [2] K. Watanabe, Y. Matsubayashi, S. Fukayama, M. Goto, K. Inui, and T. Nakano, ‘A Melody-Conditioned Lyrics Language Model’, in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, Jun. 2018, pp. 163–172. doi: 10.18653/v1/N18-1015.
- [3] X. Lu, J. Wang, B. Zhuang, S. Wang, and J. Xiao, ‘A Syllable-Structured, Contextually-Based Conditionally Generation of Chinese Lyrics’, Jun. 2019, Accessed: Nov. 12, 2021. [Online]. Available:
<https://arxiv.org/abs/1906.09322v1>
- [4] W. Lau, ‘Application of Machine Learning Model in Generating Song Lyrics’, UCLA, 2021. Accessed: Nov. 12, 2021. [Online]. Available:
<https://escholarship.org/uc/item/77d7c3hh>
- [5] A. Vaswani *et al.*, ‘Attention Is All You Need’, *arXiv:1706.03762 [cs]*, Dec. 2017, Accessed: Nov. 15, 2021. [Online]. Available:
<http://arxiv.org/abs/1706.03762>
- [6] M. Lewis *et al.*, ‘BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension’, *arXiv:1910.13461 [cs, stat]*, Oct. 2019, Accessed: Nov. 15, 2021. [Online]. Available: <http://arxiv.org/abs/1910.13461>
- [7] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, ‘BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding’, *arXiv:1810.04805 [cs]*, May 2019, Accessed: Nov. 15, 2021. [Online]. Available: <http://arxiv.org/abs/1810.04805>
- [8] H. Gill, N. Marwell, and D. (Taesoo) Lee, ‘Deep Learning in Musical Lyric Generation: An LSTM-Based Approach’, vol. 1, p. 7, 2020.
- [9] S. Gururangan *et al.*, ‘Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks’, *arXiv:2004.10964 [cs]*, May 2020, Accessed: Nov. 23, 2021. [Online]. Available: <http://arxiv.org/abs/2004.10964>
- [10] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, ‘Improving Language Understanding by Generative Pre-Training’, p. 12.
- [11] T. B. Brown *et al.*, ‘Language Models are Few-Shot Learners’, *arXiv:2005.14165 [cs]*, Jul. 2020, Accessed: Nov. 26, 2021. [Online]. Available:

- <http://arxiv.org/abs/2005.14165>
- [12] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, ‘Language Models are Unsupervised Multitask Learners’, p. 24.
- [13] S. Rothe, S. Narayan, and A. Severyn, ‘Leveraging Pre-trained Checkpoints for Sequence Generation Tasks’, *arXiv:1907.12461 [cs]*, Apr. 2020, Accessed: Nov. 12, 2021. [Online]. Available: <http://arxiv.org/abs/1907.12461>
- [14] ‘Lyrical composition in Cantopop’, *overcome man*, Jun. 16, 2011. <https://overcomeman.wordpress.com/2011/06/16/lyrical-composition-in-cantopop/> (accessed Nov. 30, 2021).
- [15] B. Shetty, ‘Natural Language Processing(NLP) for Machine Learning’, *Medium*. <https://towardsdatascience.com/natural-language-processing-nlp-for-machine-learning-d44498845d5b> (accessed Nov. 12, 2021).
- [16] D. Bahdanau, K. Cho, and Y. Bengio, ‘Neural Machine Translation by Jointly Learning to Align and Translate’, *arXiv:1409.0473 [cs, stat]*, May 2016, Accessed: Nov. 23, 2021. [Online]. Available: <http://arxiv.org/abs/1409.0473>
- [17] S. Rosalyn, ‘Nine Tones of Hell’, *China Channel*, Mar. 06, 2018. <https://chinachannel.org/2018/03/06/nine-tones-hell/> (accessed Nov. 12, 2021).
- [18] ‘OpenAI GPT2’. https://huggingface.co/transformers/model_doc/model_doc/gpt2.html (accessed Nov. 26, 2021).
- [19] A. Singh, ‘Recurrent Neural Networks : Introduction for Beginners : Introduction for Beginners’, *Analytics Vidhya*, Jun. 13, 2021. <https://www.analyticsvidhya.com/blog/2021/06/recurrent-neural-networks-introduction-for-beginners/> (accessed Nov. 22, 2021).
- [20] A. Fung, *Riding a Melodic Tide: The Development of Cantopop in Hong Kong*. Hong Kong: Subculture Press, 2009.
- [21] K. H. CHEUNG, ‘Ripples riding on waves: Cantonese tone-melody match mechanism illustrated’, in *The 18th International Conference on Yue Dialects*, JiNan University Press, 2013.
- [22] D. Jurafsky and J. H. Martin, ‘Speech and Language Processing’. <https://web.stanford.edu/~jurafsky/slp3/> (accessed Nov. 22, 2021).
- [23] E. Culurciello, ‘The fall of RNN / LSTM’, *Medium*, Jan. 10, 2019. <https://towardsdatascience.com/the-fall-of-rnn-lstm-2d1594c74ce0> (accessed Nov. 23, 2021).
- [24] J. Alammar, ‘The Illustrated Transformer’. <https://jalammar.github.io/illustrated-transformer/> (accessed Nov. 23, 2021).
- [25] D. R. Ladd and J. Kirby, ‘Tone–Melody Matching in tone–Language Singing’, *The Oxford Handbook of Language Prosody*, Dec. 31, 2020.

- <https://www.oxfordhandbooks.com/view/10.1093/oxfordhb/9780198832232.001.0001/oxfordhb-9780198832232-e-47> (accessed Dec. 01, 2021).
- [26] Hung-yi Lee, *Transformer*, (Jun. 01, 2019). Accessed: Nov. 23, 2021. [Online Video]. Available: <https://www.youtube.com/watch?v=ugWDIIOHtPA>
- [27] J. Uszkoreit, ‘Transformer: A Novel Neural Network Architecture for Language Understanding’, *Google AI Blog*. <http://ai.googleblog.com/2017/08/transformer-novel-neural-network.html> (accessed Nov. 26, 2021).
- [28] C. Olah, ‘Understanding LSTM Networks -- colah’s blog’. <https://colah.github.io/posts/2015-08-Understanding-LSTMs/> (accessed Nov. 15, 2021).
- [29] N. BM, ‘What is attention mechanism? Can I have your attention please?’, *Medium*, Jun. 20, 2021. <https://towardsdatascience.com/what-is-attention-mechanism-can-i-have-your-attention-please-3333637f2eac> (accessed Nov. 25, 2021).
- [30] ‘Common Crawl’. <https://commoncrawl.org/> (accessed Dec. 01, 2021).
- [31] ‘Wikipedia:Database download’, *Wikipedia*. Nov. 12, 2021. Accessed: Dec. 01, 2021. [Online]. Available: https://en.wikipedia.org/w/index.php?title=Wikipedia:Database_download&oldid=1054784676
- [32] ‘※ Mojim.com 魔鏡歌詞網’. <https://mojim.com/> (accessed Dec. 01, 2021).
- [33] ‘Genius | Song Lyrics & Knowledge’, *Genius*. <https://genius.com/> (accessed Dec. 01, 2021).
- [34] ‘PyCantonese: Cantonese Linguistics and NLP in Python — PyCantonese 3.3.1 documentation’. <https://pycantonese.org/> (accessed Dec. 01, 2021).
- [35] ‘BART’. https://huggingface.co/transformers/model_doc/model_doc/bart.html (accessed Dec. 01, 2021).
- [36] ‘粵語審音配詞字庫’. <https://humanum.arts.cuhk.edu.hk/Lexis/lexi-can/> (accessed Dec. 01, 2021).

Appendix

More sample lyrics generated by the four Bart models.

Input Tones:

4 3 6 5 1 1 2 2 1 5 1

5 1 1 1 3 3 5 1 1 1 3 3

2 1 1 5 1 2 2 2 1 3 1

2 1 2 2 5 5 6 3 6 1 3 5 6 3

Generated by scratch Bart model without GPT2 weight

Sample 1:

傳說什裏生中感到不理不
你的祝福說愛你不知不覺意
只需一秒經小小小歌世雖
紙之首好似你認半份不見了自覺

Sample 2:

純細膩似煙香港口中我於
你開心中結髮兩分鐘都變作
此刻開你身體只可跟送歸
彼此只相允已共痛恨跟去理做戲

Sample 3:

童世日裡跟他寫首歌也知
有一些歌唱句有一些歌唱遍
這一生有歌好玩可歌唱歌
到高到這裡有樂處亦聽著無論唱

Sample 4:

從細雨裡聽歌想想輕軟的
你聽一聲見笑臉聽聽不見細
想聽音裡聽到講聲聲線聲
感激這感染滿是笑話聲嘆美伴過

Sample 5:

寧怕沒有不需早醒的你的
也不知不覺對你心中的愛意
戀不知你不懂怎樣的結束
到不敢想你也沒法活該放下淚嗎

Sample 6:

純潔月我都一樣解心那麼
我心中的錯錯有心的不錯愛
始終不會不想怎樣失去追
可知這顆眼那是最後的愛我是錯

Sample 7:

純潔月我都一樣解心那麼
我心中的錯錯有心的不錯愛
始終不會不想怎樣失去追
可知這顆眼那是最後的愛我是錯

Sample 8:

從世代我今天討玩都也必
也必須分切要上身跟蹤氣去
到天天理不想管彼方向天
只需找到理與地發夢飛去理自己

Generated by scratch Bart model loaded GPT2 weight

Sample 1:

傻笑為了哭泣可感應你應
抱得失不要記你的一生愛錯
怎麼偏你心底這種天意飄
可不可點上眼念幼稚瀟灑語地笑

Sample 2:

寧願伴你的一起走千里親
永不知今世愛你不知傷痛卻
早知不會分手只好一切珍
早知相擁你我願世事分隔兩地再

Sample 3:

齊作用武裝多好好裝我不
老家都不怕怕你擔憂傷最愛
小姐不會驚喜這樣多要分
小夫子小女也用數十分百老日再

Sample 4:

齊作用武裝多好好裝我不
老家都不怕怕你擔憂傷最愛
小姐不會驚喜這樣多要分
小夫子小女也用數十分百老日再

Sample 5:

仍信字我多麼可使一臉傷
我都應不愛意我偏不必介意
始終的軟心比想到一切都
到一起手也有熱血亦溫意義亦愛

Sample 6:

如戰讓我的心底這一秒分
拒不出激氣派我都得多妒忌
彼不清我的感到這一寸的
可不可擺妳會就放自尊設我地帶

Sample 7:

曾太悶了今天想起都已心
也都不知結著了今天的更覺
彷彿給了一把擁緊關閉都
怎麼可否以我實際又拋棄了另愛

Sample 8:

從昨夜裡得不到可惜妳消
我的一朝快要抱得多麼痛痛
只須傷我的苦楚這刻快將
怎麼可補你我別去問孤單冷寂半

Generated by fine-tuned Bart model without GPT2 weight

Sample 1:

人各自也高歌點享心裡歌
我歌之中卻拍你跟她親笑過
可惜心裡不可比首先最初
彷彿早使妳我做錯但一切也盡過

Sample 2:

隨意聚每分鐘找到新與分
每一分鐘碰著你通宵的腳印
這一刻我不想走位先發生
否則找到了你別要用刀切抱淚印

Sample 3:

如昨日那天開始只得兩天
我身心中最愛已經超出試試
否光陰已經等好久不見天
怎麼可使我有話渴望聽著我亂說

Sample 4:

隨著是你丰姿彩起的眼睛
我的心思醉著滿天的一切祕
使風飄下憶起鼓起的笑聲
此刻想起與你共進夢經過了日記

Sample 5:

難去讓你猜得這樣的眼光
我不知不覺意也不知不覺意
這一刻我不想這夥心處居
想得這樣我也像個樂觀看似做戲

Sample 6:

如雪地老天荒島敢跟你飛
與不分之最赤柱艱辛的氣泡
這一刻已分享到喜歡氣溫
喜歡小小馬裏越唱越高貴那越秀

Sample 7:

逃過就無間飛到這都市中
無邊的方向剎那間中的百戰
只知車裡跟到這種的氣溫
怎麼想到與你渡過便足夠那陣痛

Sample 8:

人各自有歡欣點起心裡思
每天風波擦過每一天心意盼
景色跟我一夥子享福快歌
喜歡點起兩眼熱熾熱心意永沒變

Generated by fine-tuned Bart model loaded GPT2 weight

Sample 1:

來決定每一天比起都冷冰
滿身憂鬱喪氣憤傷心中怨氣
捨不得你傷口怎可將昨天
誰都想起我抱便放掉心再也未計

Sample 2:

難替代你深刻好好給你開
你輕輕的控制我心輕的控制
這不知你心好比火山更癡
想不起只有我在半夢中對你重要

Sample 3:

忘記是我一生苦守於你身
美麗的一次佈滿一生的意見
彷彿知你知否好好的轉身
想一想起你我便覺便足夠有耐性

Sample 4:

常盼望你身邊好好的晚安
你不知不覺對我多麼的歉意
怎麼收尾不想只想親切跟
懂得怎樣你我便決定不再努力去

Sample 5:

曾約定某些經片起的晚霜
我偷偷的看著你的心偷笑著
誰知心裡不懂怎樣的去追
怎麼竟這晚你熱愛地推說我願意

Sample 6:

還記得妳今天竟喜歡你麼
那一刻飛過去已經不刻意過
這一刻已不想起這歡笑麼
這刻竟想妳與別個地方對我做錯

Sample 7:

紅抹綠野風吹起好風雅姿
美得它香氣細雨一知都記載
喜歡跟有爭吵點起都變酸
喜歡好想與你伴著便出意也合意

Sample 8:

情愛若有分身彼此心已僵
我的心漸跳魄已經將得意醉
請不應有心死相戀不要分
請將這土與你命數讓分數我獨愛