# Final Report for Applying Transformer to Jyutping-to-Chinese-Character Transliteration

**Ian Quan**
ian.quan@mail.utoronto.ca

**Douglas Quan**
douglas.quan@mail.utoronto.ca

Department of Mathematical and Computational Sciences
University of Toronto
Mississauga, ON, L5L 1C6

## Abstract

This project focuses on the application of a Tramsformer model for transliteration from Jyutping to Chinese Characters. Machine transliteration plays a crucial role in natural language applications, including information retrieval and machine translation, particularly when handling with proper nouns and technical terms. Yet, there has been limited research on machine transliteration from Jyutping to Chinese Characters. Sequence-to-sequence attention-based models have shown success in transliteration tasks across various languages such as Hindi, Korean, Arabic, and even Mandarin. The Transformer, a popular model for natural language processing (NLP) tasks, has achieved state-of-the-art BLEU scores in neural machine translation without relying on RNNs or convolutions. Leveraging the exceptional performance of the Transformer, we adopt it as the fundamental architecture for the sequence-to-sequence attention-based model in Jyutping to Chinese Character transliteration.

## 1 Introduction

The Tramsformer architecture, as introduced by Vaswani et al. [2017], has emerged as a prominent solution for sequence-to-sequence translation in NLP. Leveraging the capabilities of Tramsformers, this project aims to tackle the homophonic/homonymic ambiguities inherent in transliterating Jyutping, a romanization system for Cantonese, into Chinese Characters.

Jyutping, devised by the Linguistic Society of Hong Kong, serves as the romanization system for Cantonese, aiding those unfamiliar with the native script in pronouncing the language accurately. Romanization systems, in general, play a crucial role in bridging linguistic gaps, enabling effective communication among individuals not well-versed in the intricacies of non-Latin scripts, such as Hanzi (Chinese Characters). Romanization streamlines the input of non-Latin scripts on digital devices, offering practicality by enabling users to type in Romanized characters using standard keyboards. The global adoption of Latin script elements in digital communication, driven by technological advancements in regions using the Latin script, underscores the importance of automating transliteration between different scripts.

The motivation for this project is underpinned by specific challenges faced in Cantonese NLP projects. Many interview transcripts in Cantonese are available only in Jyutping, leading to inconsistencies arising from orthographic variants and mergers in research assistant/volunteer transcriptions. To address this, the project seeks to explore methods for systematically generating character transcrip-

tions for partially annotated dataset, the aim is to enhance the consistency and reliability of these transcripts.

The significance of this project lies in its application to various fields, including speech-to-text (STT) and automatic speech recognition (ASR). The success of the Transformer model on Mandarin Chinese ASR tasks (Zhou et al. [2018]) suggests that it may also be effective for Cantonese ASR tasks. Zhou et al. [2018] also showed syllable based model with the Transformer can outperform CI-phoneme based counterpart. By developing a robust transliteration model, we aim to enhance the accuracy and efficiency of these applications, ultimately facilitating improved communication and interaction across language barriers.

Furthermore, beyond the technical challenges, this project carries cultural and societal significance. In Cantonese-speaking communities, where many interview transcripts exist solely in Jyutping, the inconsistencies in transcriptions hinder the progress of NLP projects. By systematically generating additional character transcriptions, we not only address a practical issue in NLP but also contribute to the preservation and vitality of heritage languages. Aligned with the concept of knowledge mobilization, it empowers speakers, including those less literate in written Chinese, to actively contribute to documenting their linguistic heritage, fostering a sense of ownership and pride in preserving cultural richness.

## 2 Background and Related Work

### 2.1 Related Work

Sequence-to-sequence attention-based models have shown very encouraging results on transliteration tasks on different languages. In Singh and Bansal [2021], various architectures of encoder-decoder models were used for transliteration tasks on Hindi and Punjabi languages and gave state-of-the-art-results. Our approach draws inspiration from prior research that uses a deep-learning approach for transliteration tasks. Finch et al. [2016] introduced an ensemble Bi-Directional LSTM model exclusively designed for Neural Machine Transliteration. The model combines multiple networks through linear interpolation, assigning equal weights to the probability distributions generated by these networks across the target vocabulary during beam search decoding. Although the model yielded satisfactory results for Indian languages, it is a very complex model and relies on RNNs, which suffer form the vanishing gradient problem. Consequently, its performance diminishes in handling longer sequences or sentences where contextual information is crucial, even though the LSTM model is effective for individual words.Wu et al. [2020] shows that with a large enough batch size, the Tramsformer does indeed outperforms recurrent models on character-level tasks.

### 2.2 Homophone Ambiguity

Transliteration, the act of mapping a word from the orthographic system of one language to another, is directed by the pronunciation in the source and target languages. In this project, we endeavour to automate transliteration from Jyutping to Chinese Character, which is also know as *Hanzi*. Jyutping is a commonly used romanization system for Cantonese, and it represents the sounds of the language. For example, the Jyutping *nei5 hou2* can be transliterated to the Chinese Characters 你好. However, there is not a one-to-one mapping between Jyutping and Chinese Characters, as multiple characters can correspond to the same Jyutping representation (see table 4), which notoriously known as the Homophone Ambiguity. Kwong [2009] suggests that a considerable part of homophones used in the transliterations could be distinguished by tones. Thus, disambiguation can be performed as words are typically constructed using multiple mono-syllabic morphemes [1] that can be easily resolved in context. This aligns with our hypothesis that understanding the context as well as the sound associated with the written words using a Tramsformer model helps solve the problem of Homophone Ambiguity during transliteration.

[1] In Chinese, morpheme can be divided into mono-syllabic morpheme and multi-syllabic morpheme.The mono-syllabic morpheme composed of only one syllable, which is the basic form of the morpheme in Chinese language. It is a syllable when being read while a character when being written such as 人, 去, 我, 往, 其, 很 etc. Mono-syllabic morpheme is commonly used for word formation. In contrast, multi-syllabic morpheme is just one syllable but two characters when being written but have only one meaning such as 秋千, 蜘蛛, 激光, 雞蛋. Keats-School [2018]

Table 1: Example of Cantonese homophone characters

| Jyutping | Homophone Characters |
|---|---|
| sik1 | 識, 適, 悉, 色, 式, 釋, 息, 晰, 惜 |
| si1 | 嘶, 司, 斯, 思, 私, 師, 獅, 詩, 屍 |
| jyun4 | 原, 完, 員, 園, 圓, 緣, 元, 源 |

# 3 Data

## 3.1 Data Sourcing

Linguistic resources suitable for Cantonese NLP applications are quite limited, particularly when it comes to the availability of Cantonese-Jyutping parallel corpora. Although there are varying sizes of Cantonese-English corpora, such as the Hong Kong Hansard and the Hong Kong Laws Parallel Text, they all lack Cantonese-Jyutping pairs and require further pre-processing in order for language models to efficiently learn linguistic features such as word segments, part-of-speech (POS) or named-entity.

We have chosen 3 conversational corpora and 1 lexical wordlist as our datasets for this project. Conversational data includes informal language, colloquialisms, and slang. Including conversational data helps the model handle real-world scenarios where people might use informal language. A larger proportion of conversational data will be trained on since the model primarily designed for conversational application such as ASR and speech transcriptions. On the other hand, A lexical wordlist contains a wide range of words, including domain-specific and rare terms. Training on such data ensures better coverage across different topics and vocabulary. This can potential solve problems often encountered in Word-based neural network models, such as limited vocabulary size, rare word and out-of-vocabulary (OOV) problems.

To our knowledge, the Hambaanglaang Storybooks(unpublished), CantoMap, the Cantonese MapTask corpus (Winterstein et al. [2020]) and The Hong Kong Cantonese Corpus (Luke and Wong [2015]) are the few existing publicly available Cantonese-Jyutping parallel corpora. In addition, words.hk Lau et al. [2022] is chosen to be our lexical dataset [2].

## 3.2 Data Overview

Hambaanglaang Storybooks, initiated by Viveik Mohan Saigal and Dr. Chaak Ming Lau, addresses the need for foundational Cantonese learning resources. The project aims to enhance Cantonese proficiency by providing storybooks for learners of all ages. The Hambaanglaang Storybooks dataset consists of the narration, including storyline and dialogue, of 235 storybooks.

CantoMap, the Cantonese MapTask corpus is a collection of recordings of the MapTask task in contemporary spoken Hong Kong Cantonese. The corpus aims at providing an additional resource for the study of modern Cantonese. The corpus contains a total of 768 minutes of recordings and transcripts of 40 speakers.

The Hong Kong Cantonese Corpus (HKCanCor) was collected from transcribed conversations that were recorded between March 1997 and August 1998. It contains recordings of spontaneous speech (51 texts) and radio programmes (42 texts), which involve 2 to 4 speakers, with 1 text of monologue.

words.hk(粵典) is a large-scale Cantonese lexicon project, which aims to create an extensive and sustainable Cantonese dictionary. words.hk word list(粵典詞表) is a lexical database which contains lemmas in Chinese Character and Jyutping of all entries recorded in the dictionary.

All datasets mentioned above are summarized in table 5.

[2]Data licences of the datasets:
Hambaanglaang storybooks and HKCanCor are released under the CC BY 4.0 license.
CantoMap is released under the GNU General Public License v3.0.
words.hk word list is released under Open Data Commons Open Database License (ODbL) v1.0

# 4 Model Architecture

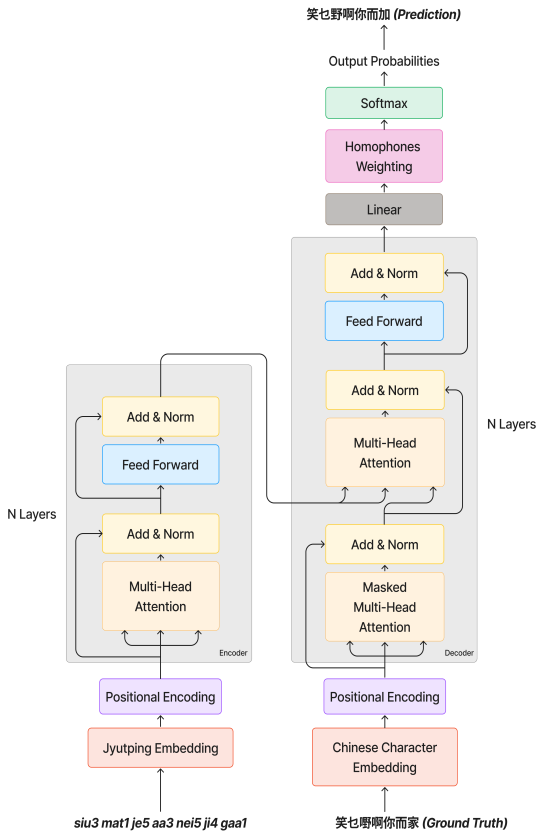


Figure 1: Transformer Model Architecture.

The architecture of the Transformer model is shown in Figure 1. We implemented a customized variant of the sequence-to-sequence attention-based Transformer model proposed by Vaswani et al. [2017], augmented with the inclusion of a Homophones Weighting Layer. At a high level, the encoder of our model maps an input sequence — in this context, a Jyutping sentence denoted as $\mathbf{x} = [x_1, ..., x_n]$ — to a sequence of context-aware representations $\mathbf{z} = [z_1, ..., z_n]$, embedded with positional encoding and self-attention mechanisms. The decoder, operating on this context-aware representation $\mathbf{z}$, generates an output sequence $\mathbf{y} = [y_1, ..., y_n]$, in this case a translated Chinese Character sentence, one element at a time.

Note that the lengths of both the input and output sequences are identical. This coherence ensures a one-to-one correspondence between Jyutping and translated Chinese Characters throughout the sequence. Below is a concise overview of the key components of our Transformer model architecture:

## 4.1 Word Embedding

Before passing the data into the encoder or decoder, we need to first convert the Jyutping and Chinese Characters into vectors.

**Jyutping Embedding:** As mentioned before, there are no publicly available pre-trained word embeddings for Jyutping. For this reason, we had to build and train our own word embedding model from one-hot encodings we prepared for Jyutping for this project.

**Chinese Character Embedding:** Previous work Edunov et al. [2019] has shown that models can be improved by incorporating pre-trained word embedding encoder. To convert Chinese Characters into word vector, we used a pre-trained word embedding model bart-base-cantonese created by Ayaka [2022]. This is a model that performs second-stage pre-training with Cantonese data on the BART base Chinese model.

### 4.2 Positional Encoding

Positional encoding incorporates positional information for each token within the sequence into the embedding vectors. Analogous to translation tasks, we utilize the model's capacity to learn sequential relationships within the sentence. This capability proves instrumental in addressing the homophone ambiguity problem our task, enabling the model to distinguish between homophones by leveraging their respective positions in the sequence. The formula for positional encoding we used:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

$$PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{model}}}\right)$$

where $d_{model}$ denotes the model dimension, we set $d_{model} = 512$ for this project.

### 4.3 Encoder

The encoder module is a stacked, non-autoregressive component designed to capture contextual relationships among words in the input Jyutping sentence. The encoder is composed of a stack of N identical layers, we used 5 layers in this project. Each layer has two sub-layers: a multi-head attention (MHA) layer and a position-wise fully-connected feed-forward network (FFN). Residual connections surround each of these sub-layers, followed by layer normalization.

**Multi Head Attention(MHA)** computes attention weights and contextual vectors from the query, key and value vectors, all of which are learned during training. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. The equation of a single attention can be expressed as follows Vaswani et al. [2017]:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Instead of utilizing a single attention function, the Transformer adopts multi-head attention (MHA), a technique that involves projecting the queries, keys, and values $h$ times, we set $h = 8$ in this project, using distinct learned linear projections to dimensions $d_k$, $d_k$, and $d_v$. For each of these projected versions of queries, keys, and values, the attention function is independently applied in parallel, producing output values in a $d_v$-dimensional space. These outputs are then concatenated and subjected to another projection, ultimately yielding the final values. The equation of MHA can be expressed as follows Vaswani et al. [2017]:

$$MultiHead(Q, K, V) = Concat(head_1, ..., head_h)W^O$$

$$where \quad head_i = Attention(QW_i^Q, KW_i^K, VW_i^V)$$

Given that the projections are matrices $W_i^Q, W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_v}$, $W^O \in \mathbb{R}^{hd_v \times d_{model}}$, $h$ is the number of attention heads, and $d_{model}$ is the model dimension. A detailed decomposition of MHA can be viewed in Figure 2.

**Fully-Connected Feed-forward Network(FFN)** transforms attention/contextual vectors into a format more amenable for the subsequent layer in our model architecture.

The stacking of multiple encoder modules amplifies the neural network's capacity and enhance performance. By increasing layer depth, the model can simultaneously learn in multiple semantic levels of feature space.

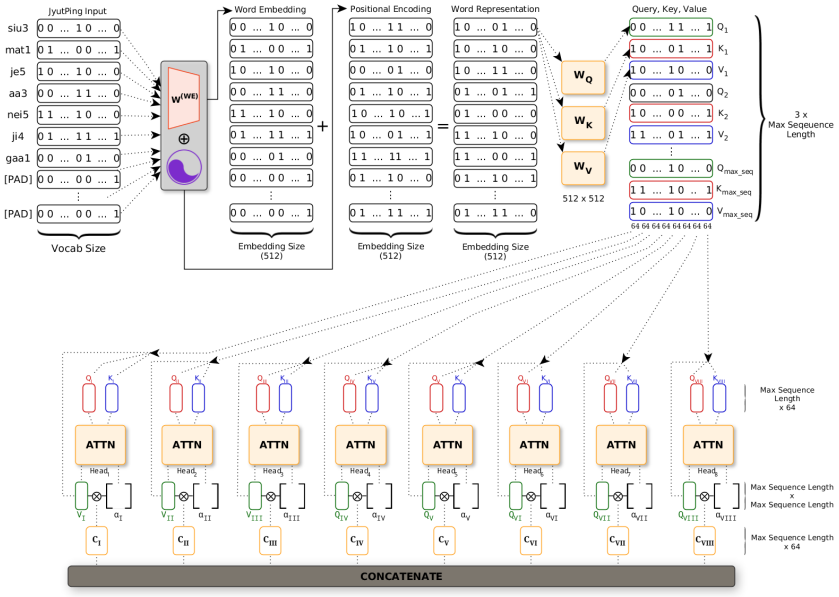


Figure 2: Multi-Head Attention.

### 4.4 Decoder

The decoder, on the other hand, is a stacked, autoregressive module responsible for capturing interactions between Jyutping and Chinese Characters. Similar to the encoder, the decoder stacks MHA and position-wise FFN in each layer. However, the decoder introduces a third sub-layer to perform MHA over the output of the encoder stack. To maintain the auto-regressive property and prevent leftward information flow, the self-attention sub-layers in the decoder mask out all values corresponding to illegal connections.

The decoder's output consists of attentive vectors, each representing relationships with words between the Jyutping and Chinese Character. The output is then passed through a linear layer, followed by a homophones weighting Layer and softmax activation, to obtain probability distributions over the possible Chinese Characters.

### 4.5 Homophones Weighting (HW)

Unlike translation tasks, where translated sentences may exhibit variations in length and structural composition, transliteration primarily involves the mapping of words from one writing system to another. In our case, the task is to map Jyutping to Chinese Characters. To enhance the model's performance in capturing homophonic relationship, we introduce an additional Homophones Weighting layer, following the linear layer output from the decoder.

The decoder produces a set of logits for each class, with each class representing a potential Chinese Character. Consequently, the size of the logit vector aligns with the total number of possible Chinese Characters that the model can predict. The Homophones Weighting layer introduces a weighting mechanism to emphasize predictions that correspond to the homophones of the input character. For instance, when the input Jyutping is *gat1*, the logits associated with all of its homophonic Chinese Characters (吉, 拮, 揭, 桔) are multiplied by a hyperparameter denoted as $\gamma$. This process resembles a post-processing attention mechanism applied to the output probabilities before finalizing the prediction. The output of this layer can be expressed as:

$$weighted\_logits = logits \times \Gamma$$

Here, $\Gamma$ represents the hyperparameter matrix containing $\gamma$ values associated with each class. In this project, we employ $\gamma = 1.1$.

## 5 Results

In evaluating our model's performance, we employed a systematic approach, starting with a baseline configuration and iteratively refining it through the implementation of homophone weighting and hyperparameter tuning with Ray Tune [5]. All models are trained for 30 epochs.

The hyperparameters chosen for the baseline model are informed by the objective of establishing a model with reasonable performance and provides a solid foundation for subsequent refinements. The testing accuracy of 51% reflects the model's initial capability.

Upon incorporating the Homophone Weighting technique during tuning, an additional hyperparameter, homophone weight ($\gamma$) of 1.1, was introduced to the model's configuration. This refinement led to a notable performance improvement, elevating the accuracy to 78% on the testing set.

Subsequently, hyperparameter tuning further optimized the model, achieving a remarkable accuracy of 94% on the testing set.

The validation accuracy and training loss during the tuning process can be viewed in Figure 3 and Figure 4.

Table 2: Hyperparameter settings on different models

| Model | $d_{\text{model}}$ | hidden_size | num_heads | dropout | num_layers | batch_size | lr | $\gamma$ |
|---|---|---|---|---|---|---|---|---|
| Baseline | 512 | 2048 | 6 | 0.1 | 3 | 32 | 0.0001 | / |
| Baseline+HW | 512 | 2048 | 6 | 0.1 | 3 | 32 | 0.0001 | 1.2 |
| Tuned+HW | **512** | **2048** | **8** | **0.2** | **5** | **64** | **0.0002** | **1.1** |

Table 3: Test performance of the tuned model against different models

| Model | Test Accuracy | Training Loss |
|---|---|---|
| Baseline | 0.512 | 0.2538 |
| Baseline+HW | 0.783 | 0.2334 |
| Tuned+HW | **0.948** | **0.0684** |
| Vanilla RNN | 0.758 | 0.1425 |

## 6 Discussion

Along the optimal set of hyperparameter values found after tuning, we perform our empricial evaluation on the three datasets Hambaanglang, CantoMap and HKCanCor with 18207 training examples and 4551 test samples. Comparing the test accuracy across all model, see table 3, it is clear that the tuned model with HW outperforms other model significantly and obtian a very promising result with a 94% test accuracy.

The integration of Homophone Weighting stands out as a pivotal refinement step. The introduction of the homophone weight ($\gamma$) at 1.1 manifests a deliberate emphasis on homophones during training. The model's enhanced sensitivity to homophonic relationships serves as clear evidence of the effectiveness of this modification.

We also conducted a comparative analysis of the Transformer model and a vanilla Recurrent Neural Network (RNN) to assess their performance in addressing the challenges related to Homophone Ambiguity. The results indicate that the Transformer model surpasses the vanilla RNN, Table 3, supporting our hypothesis that the RNN's limitations in capturing long-range dependencies and maintaining contextual information over sequences hinder its ability to effectively distinguish between homophones in intricate linguistic contexts. In contrast, the Transformer's multi-head attention mechanism enables parallel processing of information. This feature allows the model to concurrently consider various aspects of the input sequence, facilitating a more robust understanding of the contextual relationships. The superior performance of the Transformer underscores the importance of parallelism and contextual awareness in handling Homophone Ambiguity.

## 7 Limitations

While our Tramsformer model for Jyutping-to-Chinese Character transliteration has demonstrated promising results, several limitations should be acknowledged:

**Difficulty in Handling Specific Terms and Lexicons:** Transliterating specific terms, names, and phrases poses a significant challenge. The nuances of Cantonese pronunciation and the inherent variability in representing specific terms in Jyutping may result in transliteration errors for domain-specific vocabulary. The model may struggle with transliterating words or terms that are not present in the training data, leading to out-of-vocabulary challenges for unfamiliar or domain-specific terminology.

From the example below, The model's transliteration of the Jyutping term "*gaai3 ngoi6 kau4*" as "介外球" instead of the expected "界外球" reveals a limitation in accurately rendering specific lexicons. In this context, "gaai3 ngoi6 kau4" translates to "throw-in" (out-of-bounds). It exemplifies the model's current incapacity in transcribing domain-specific terms.

Table 4: Example of current model's incapacity in transcribing domain-specific terms

| | |
|---|---|
| Jyutping | gaai3 ngoi6 kau4 hoi1 faan1 mai6 daai6 gaa1 nou5 lik6 zaang1 faan1 |
| Character Transliteration | 界外球開返咪大家努力爭返 |
| Model Character Prediction | 介外球開返咪大家努力爭返 |

**Insufficiency in Data:** Cantonese, especially Jyutping is a low-resource languages in the context of natural language processing. This presents challenges related to data scarcity. Limited availability of diverse and comprehensive datasets may hinder the model's ability to generalize well, particularly for uncommon words, names, or phrases.

## 8 Ethical Considerations

When dealing with conversation transcriptions from recordings of interviews with Cantonese speakers, ensuring privacy, confidentiality, and obtaining informed consent are critical considerations.

Interview transcripts may contain sensitive information about individuals, including personal stories, opinions, or details about their lives. It's crucial to identify and safeguard such sensitive content to protect the privacy of the participants.

Ensure anonymization techniques are used in our chosen dataset to remove or replace personally identifiable information (PII) from the transcripts. This may include names, addresses, or any other details that could potentially reveal the identity of the participants.

## 9 Conclusion

We presented a Tramsformer model tailored for the task of transliterating from Jyutping to Chinese Characters, with a particular focus on addressing the issue of homophone ambiguity. Our Tramsformer model represents a significant step forward in addressing the complexities of Jyutping transliteration by leveraging techniques such as attention mechanisms, positional encoding, and the homophone weighting layer, the model has demonstrated promising accuracy in capturing the nuanced relationship between characters in a sentence.

However, as highlighted in the limitations section, challenges remain. Future work should focus on addressing these challenges and exploring potential enhancements to the model's robustness. Strategies to refine data quality, handle domain-specific terms, and improve generalization across dialectal variations will be vital for the continued research.

# Appendix

Table 5: Summary of datasets

| | Hambaanglaang Storybooks | CantoMap MapTask | HKCanCor | words.hk |
|---|---|---|---|---|
| Number of utterances: | 9078 | 11350 | 10801 | 55593 |
| Number of words: | 140189 | 82395 | 165211 | 137339 |
| Average words per utterance: | 15.443 | 7.259 | 15.296 | 2.47 |
| Longest sentence size: | 96 | 357 | 718 | 13 |
| Number of files: | 235 | 40 | 93 | / |

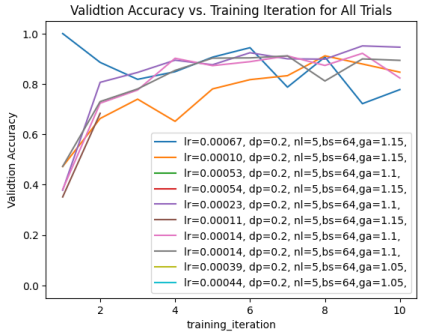


Figure 3: Validation accuracy during hyperparameter tuning.

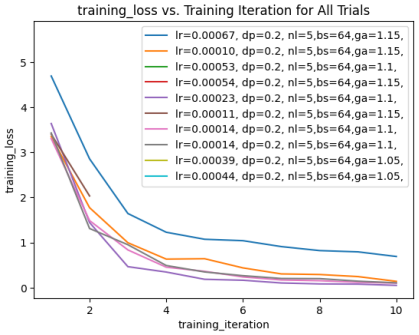


Figure 4: Training loss during hyperparameter tuning.

| Trial name | status | d_model | ffn_hidden | num_heads | drop_prob | num_layers | batch_size | lr | gamma | iter | total time (s) | training_loss | accuracy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ray_train_d493c_00000 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000667603 | 1.15 | 10 | 711.265 | 0.690964 | 0.777475 |
| ray_train_d493c_00001 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000103222 | 1.15 | 10 | 789.294 | 0.139936 | 0.846633 |
| ray_train_d493c_00002 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000532714 | 1.1 | 1 | 67.3867 | 5.30992 | 1 |
| ray_train_d493c_00003 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000535209 | 1.15 | 1 | 69.0418 | 4.96176 | 0.556818 |
| ray_train_d493c_00004 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000225804 | 1.1 | 10 | 783.727 | 0.0504341 | 0.945971 |
| ray_train_d493c_00005 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000110149 | 1.15 | 2 | 156.04 | 2.033 | 0.683111 |
| ray_train_d493c_00006 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000143035 | 1.1 | 10 | 781.05 | 0.10039 | 0.823453 |
| ray_train_d493c_00007 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000139641 | 1.1 | 10 | 784.699 | 0.107108 | 0.893659 |
| ray_train_d493c_00008 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000387438 | 1.05 | 1 | 75.4635 | 4.49154 | 0.0281666 |
| ray_train_d493c_00009 | TERMINATED | 512 | 2048 | 8 | 0.2 | 5 | 64 | 0.000442779 | 1.05 | 1 | 68.4965 | 5.6345 | 0 |

Figure 5: Ray Tune result.

# References


Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Shiyu Zhou, Linhao Dong, Shuang Xu, and Bo Xu. Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese, 2018.

Aryan Singh and Jhalak Bansal. Neural machine transliteration of indian languages. In *2021 4th International Conference on Computing and Communications Technologies (ICCCT)*, pages 91–96, 2021. doi: 10.1109/ICCCT53315.2021.9711806.

Andrew Finch, Lemao Liu, Xiaolin Wang, and Eiichiro Sumita. Target-bidirectional neural models for machine transliteration. In *Proceedings of the sixth named entity workshop*, pages 78–82, 2016.

Shijie Wu, Ryan Cotterell, and Mans Hulden. Applying the transformer to character-level transduction. *CoRR*, abs/2005.10213, 2020. URL `https://arxiv.org/abs/2005.10213`.

Olivia OY Kwong. Homophones and tonal patterns in english-chinese transliteration. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, pages 21–24, 2009.

Keats-School. Monosyllabic morphemes and multi-syllabic morphemes in mandarin, 2018. URL https://keatschinese.com/china-culture-resources/monosyllabic-morphemes-and-multi-syllabic-morphemes-in-mandarin/#:~:text=The%20monosyllabic%20morpheme%20is%20the,%E3%80%81%E8%87%AA%E3%80%81%E4%BB%A5%2C%20etc.

Grégoire Winterstein, Carmen Tang, and Regine Lai. CantoMap: a Hong Kong Cantonese MapTask corpus. In Nicoletta Calzolari, Frédéric Béchet, Philippe Blache, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 2906–2913, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL https://aclanthology.org/2020.lrec-1.355.

K. K. Luke and May L.Y. Wong. The hong kong cantonese corpus: Design and uses journal of chinese linguistics. 2015.

Chaak-ming Lau, Grace Wing-yan Chan, Raymond Ka-wai Tse, and Lilian Suet-ying Chan. Words.hk: A comprehensive Cantonese dictionary dataset with definitions, translations and transliterated examples. In Jonne Sälevä and Constantine Lignos, editors, *Proceedings of the Workshop on Dataset Creation for Lower-Resourced Languages within the 13th Language Resources and Evaluation Conference*, pages 53–62, Marseille, France, June 2022. European Language Resources Association. URL https://aclanthology.org/2022.dclrl-1.7.

Sergey Edunov, Alexei Baevski, and Michael Auli. Pre-trained language model representations for language generation, 2019.

Ayaka. huggingface.co bart-base-cantonese, 2022. URL https://huggingface.co/Ayaka/bart-base-cantonese#bart-base-cantonese.