



มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขต ศิริราช

คณะ วิศวกรรมศาสตร์ สาขา วิศวกรรมคอมพิวเตอร์

รายงานโครงงานเรื่อง

Generate Story by 5 words

จัดทำโดย

นาย ธนวัฒน์ พิตตินันท์ รหัสนิสิต 6530300236

คณะ วิศวกรรมศาสตร์ สาขา วิศวกรรมคอมพิวเตอร์

อาจารย์ที่ปรึกษา

ดร.ฐนียา สัตยพานิช

ภาคการศึกษา เทอม 1 ปีการศึกษา 2567

มหาวิทยาลัยเกษตรศาสตร์ วิทยาเขต ศิริราช

ปีพุทธศักราช 2567

## ที่มาและความสำคัญของปัญหา

การเรียนรู้เชิงลึก (Deep Learning) ได้รับความสนใจอย่างมากในปัจจุบัน โดยเฉพาะในด้านการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) การพัฒนาโมเดลภาษา (Language Models) ที่สามารถสร้างสรรค์ข้อความเชิงสร้างสรรค์ เช่น บทความ ขำ และเรื่องราวต่าง ๆ ได้อย่างคล่องแคล่วไม่ต่างจากมนุษย์ ถือเป็นความก้าวหน้าทางเทคโนโลยีที่มีศักยภาพสูง

GPT-2 (Generative Pretrained Transformer 2) ซึ่งเป็นหนึ่งในโมเดลภาษาแบบ Transformer ที่พัฒนาโดย OpenAI มีความสามารถในการสร้างข้อความโดยอ้างอิงจากข้อมูลที่ได้รับ เช่น ชุดคำ 5 คำที่กำหนด โค้ดนี้จึงนำ GPT-2 มาใช้ในการสร้างเรื่องราวตามคำ 5 คำที่ผู้ใช้ป้อน เพื่อทดสอบความสามารถของโมเดลในการสร้างเนื้อหาที่สอดคล้องและเข้าใจได้

## ปัญหา

การสร้างเรื่องราวที่มีความสมเหตุสมผลและตรงกับคำหลัก (keywords) ที่ผู้ใช้ป้อนเป็นความท้าทายสำคัญของการใช้โมเดลภาษา เนื่องจากโมเดลอาจไม่สามารถสร้างเนื้อหาที่เชื่อมโยงกับคำที่กำหนดได้อย่างสมบูรณ์ หรืออาจสร้างข้อความที่ซ้ำซ้อนหรือขาดความสมเหตุสมผล ดังนั้นการพัฒนาและปรับปรุงโมเดลจึงจำเป็นต้องใช้เทคนิคการฝึกฝนและประเมินประสิทธิภาพของโมเดลเพื่อให้ได้ผลลัพธ์ที่มีคุณภาพสูงที่สุด

โดยเฉพาะอย่างยิ่งในการประเมินผลการทำงานของโมเดล การใช้ BLEU score ซึ่งเป็นเครื่องมือวัดความใกล้เคียงระหว่างข้อความที่โมเดลสร้างขึ้นกับข้อความต้นฉบับ จะช่วยให้สามารถวัดผลการทำงานของโมเดลได้อย่างมีประสิทธิภาพและนำมาปรับปรุงโมเดลให้ดียิ่งขึ้น

## วิธีการเก็บรวบรวมข้อมูล

ในการทดสอบความสามารถของโมเดล GPT-2 ในการสร้างเรื่องราวจากคำหลักที่กำหนด โครงการนี้ใช้ข้อมูลที่เกี่ยวข้องกับคำสำคัญและเรื่องราวที่เกี่ยวข้องจากเว็บไซต์ต่างๆตามที่ได้กล่าวไปหัวข้อ [ขอบคุณข้อมูลที่น่ามาประกอบ Dataset] ในหน้าท้ายๆ โดยข้อมูลจะถูกรวมไว้ใน Dataset และแยกข้อมูลที่เก็บรวบรวมประกอบด้วย 2 ส่วนหลัก ได้แก่ ข้อมูลสำหรับการฝึกโมเดล (Training Data) และข้อมูลสำหรับการทดสอบโมเดล (Test Data) โดยมีวิธีการดังต่อไปนี้:

### 1. ข้อมูลสำหรับการฝึกโมเดล (Training Data)

ข้อมูลนี้ถูกรวบรวมจากชุดข้อมูลที่ประกอบด้วยคำ 5 คำ (word1, word2, word3, word4, word5) และเรื่องราวที่เกี่ยวข้องกับคำเหล่านั้น ข้อมูลเหล่านี้นำมาจัดเก็บในรูปแบบไฟล์ CSV โดยมีการรวมคำ 5 คำและเรื่องราวเข้าด้วยกันในรูปแบบข้อความเดียวเพื่อใช้เป็นอินพุตสำหรับการฝึกโมเดล GPT-2

ตัวอย่างข้อมูลการฝึก:

- Word: "apple", "banana", "orange", "grape", "watermelon"
- Story: "In the summer, the fruit market was full of fresh apples, bananas, and oranges. Grapes and watermelons were the favorites among customers."

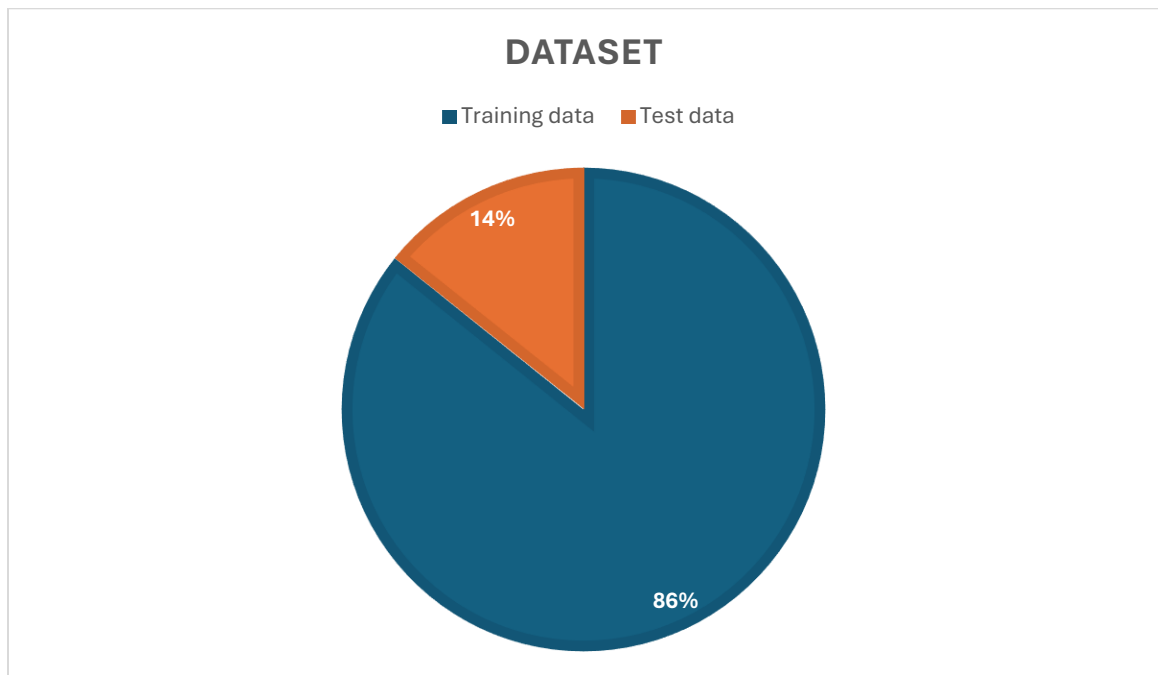
### 2. ข้อมูลสำหรับการทดสอบโมเดล (Test Data)

ข้อมูลการทดสอบจะประกอบด้วยคำ 5 คำเช่นเดียวกัน แต่ไม่มีเรื่องราว เพื่อใช้ในการประเมินผลการสร้างเรื่องราวของโมเดล ข้อมูลนี้จะถูกรวบรวมในรูปแบบไฟล์ CSV ที่มีเฉพาะคำหลัก โดยโมเดลจะต้องสร้างเรื่องราวที่มีความสัมพันธ์กับคำเหล่านี้

### 3. การจัดการข้อมูล

หลังจากที่ได้ข้อมูลมาแล้ว ข้อมูลจะถูกนำไปจัดรูปแบบในลักษณะที่เหมาะสมสำหรับการใช้งานกับโมเดล GPT-2 โดยจะทำการรวมคำหลักและเรื่องราวเข้าด้วยกันในกรณีของข้อมูลฝึก และเก็บคำหลักแยกต่างหากในกรณีของข้อมูลทดสอบ

## สัดส่วนการแบ่ง Dataset เป็น Training set และ Test set



- Dataset มีทั้งหมด 1131 ชุด
- Testset มีทั้งหมด 161 ชุด
- Trainset มีทั้งหมด 970 ชุด

โดยผู้จัดทำได้เขียนโค้ดชุดนี้ทำงานเกี่ยวกับการฝึก (fine-tuning) โมเดลภาษาของ GPT-2 เพื่อสร้างเรื่องราว (story) ตามคำ 5 คำที่ผู้ใช้ป้อน และยังทำการประเมินผลของโมเดลด้วยการคำนวณค่า BLEU score ซึ่งเป็นมาตรวัดสำหรับการวัดความใกล้เคียงระหว่างข้อความที่โมเดลสร้างขึ้นกับข้อความต้นแบบที่ถูกต้อง โค้ดนี้สามารถแบ่งการทำงานออกเป็นส่วนตัวต่าง ๆ ได้ดังนี้:

#### 1. การนำเข้า Libraries ที่จำเป็น

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import torch
import random
import nltk
from transformers import GPT2LMHeadModel, GPT2Tokenizer, TextDataset, DataCollatorForLanguageModeling
from transformers import Trainer, TrainingArguments, EarlyStoppingCallback, TrainerCallback
from nltk.translate.bleu_score import sentence_bleu
from nltk.tokenize import word_tokenize
from torch.optim import AdamW

nltk.download('punkt')
```

- ใช้ pandas สำหรับจัดการข้อมูลในรูปแบบตาราง, numpy สำหรับการคำนวณเชิงตัวเลข, และ matplotlib สำหรับการสร้างกราฟ
- ใช้ torch สำหรับการทำงานที่เกี่ยวกับ Deep Learning เช่น การกำหนด seed และการใช้โมเดล
- ใช้ nltk สำหรับการคำนวณค่า BLEU score ซึ่งช่วยประเมินคุณภาพของเรื่องราวที่โมเดลสร้างขึ้น
- ส่วนที่เหลือจะเป็นการนำเข้าโมเดล GPT-2 และเครื่องมือที่ช่วยในการปรับแต่งโมเดล (fine-tuning) เช่น Trainer และ TrainingArguments

## 2. การกำหนด seed

```
# Set seeds
def set_seeds(seed=42):
    random.seed(seed)
    np.random.seed(seed)
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    torch.backends.cudnn.deterministic = True
    torch.backends.cudnn.benchmark = False
set_seeds()
```

การตั้งค่า seed เพื่อให้การทำงานซ้ำกันได้ ซึ่งจะทำให้ผลลัพธ์ของการรันโค้ดในแต่ละครั้งเป็นไปในทิศทางเดียวกัน

## 3. การโหลดข้อมูล

```
#Load File
def load_train_data(file_path):
    df = pd.read_csv(file_path)
    df['combined'] = df.apply(lambda row: f"Input: {row['word1']} {row['word2']} {row['word3']} {row['word4']} {row['word5']}\nStory: {row['story']}", axis=1)
    return df['combined'].tolist()

def load_test_data(file_path):
    df = pd.read_csv(file_path)
    return df[['word1', 'word2', 'word3', 'word4', 'word5']].values.tolist()

def load_dataset(file_path):
    return pd.read_csv(file_path)
```

- ฟังก์ชัน load\_train\_data คือ โหลดข้อมูลจากไฟล์ CSV และรวมคำ 5 คำ พร้อมกับเรื่องราวเข้าด้วยกันเป็นข้อความเดียวที่ใช้สำหรับการฝึกโมเดล
- ฟังก์ชัน load\_test\_data คือ โหลดข้อมูลชุดทดสอบ โดยจะดึงเฉพาะคำ 5 คำจากแต่ละแถวเท่านั้น
- ฟังก์ชัน load\_dataset คือ โหลด dataset ทั้งหมดจากไฟล์ CSV

#### 4. การสร้างกราฟ loss/steps

```
#graph loss/steps
class PlottingCallback(TrainerCallback):
    def __init__(self):
        self.train_losses = []
        self.train_perplexities = []
        self.steps = []

    def on_log(self, args, state, control, logs=None, **kwargs):
        print(f"Step: {state.global_step}")
        print(f"Loss: {logs.get('loss')}")
        print("-" * 50)
        if logs is not None and state.is_local_process_zero:
            step = state.global_step
            loss = logs.get('loss')

            if loss is not None:
                self.train_losses.append((step, loss))

            self.steps.append(step)
```

คลาสนี้สร้าง callback เพื่อบันทึกค่า loss ที่เกิดขึ้นในระหว่างการฝึก และนำมาแสดงผลเป็นกราฟในภายหลัง

#### 5. การสร้างเรื่องราว (Story Generation)

```
def generate_story(model, tokenizer, words):
    input_text = f"Words: {' '.join(words)}\nStory:"
    input_ids = tokenizer.encode(input_text, return_tensors='pt')

    output = model.generate(
        input_ids,
        max_length=100,
        num_return_sequences=1,
        no_repeat_ngram_size=3,
        top_k=50,
        top_p=0.95,
        temperature=0.7,
        do_sample=True
    )
    generated_text = tokenizer.decode(output[0], skip_special_tokens=True)
    # story = generated_text.split('Story:')[1].strip()
    story = generated_text.split('Story:')[1].strip().replace('<|endoftext|>', '').strip()

    return story
```

ฟังก์ชันนี้ใช้โมเดล GPT-2 ที่ได้รับการฝึกแล้วเพื่อสร้างเรื่องราวตามคำ 5 คำที่ป้อนเข้าไป โดยใช้เทคนิคต่าง ๆ เช่น top-k และ top-p sampling เพื่อลดความซ้ำซ้อนในเรื่องราวที่สร้างขึ้น

## 6. การฝึกโมเดล (Fine-Tuning)

```
def fine_tune_model(data):  
    model = GPT2LMHeadModel.from_pretrained('gpt2')  
    tokenizer = GPT2Tokenizer.from_pretrained('gpt2')  
    tokenizer.pad_token = tokenizer.eos_token  
    model.resize_token_embeddings(len(tokenizer))  
  
    with open('temp_train_data.txt', 'w', encoding='utf-8') as f:  
        for item in data:  
            f.write(f"{item}\n")  
  
    dataset = TextDataset(  
        tokenizer=tokenizer,  
        file_path="temp_train_data.txt",  
        block_size=128  
    )  
  
    data_collator = DataCollatorForLanguageModeling(  
        tokenizer=tokenizer, mlm=False  
    )  
  
    training_args = TrainingArguments(  
        output_dir="./results",  
        overwrite_output_dir=True,  
        num_train_epochs=4,  
        per_device_train_batch_size=2,  
        gradient_accumulation_steps=4,  
        save_steps=500,  
        save_total_limit=2,  
        eval_strategy="steps",  
        eval_steps=500,  
        load_best_model_at_end=True,  
        logging_dir='./logs',  
        logging_steps=1,  
    )  
  
    optimizer = AdamW(model.parameters(), lr=5e-5)  
  
    early_stopping_callback = EarlyStoppingCallback(early_stopping_patience=3)  
    plotting_callback = PlottingCallback()  
  
    trainer = Trainer(  
        model=model,  
        args=training_args,  
        data_collator=data_collator,  
        train_dataset=dataset,  
        optimizers=(optimizer, None),  
        callbacks=[early_stopping_callback, plotting_callback],  
    )  
  
    trainer.train()
```

1  
2  
3  
4  
5  
6  
7  
8  
9



## ขั้นตอนการสร้างโมเดล

### 1. โหลดโมเดลและโทเคนไนเซอร์ (Tokenizer)

บรรทัดนี้ใช้ GPT2LMHeadModel และ GPT2Tokenizer เพื่อโหลดโมเดล GPT-2 และโทเคนไนเซอร์ที่ถูกฝึกมาแล้วจาก Hugging Face (ผ่าน from\_pretrained ซึ่งจะโหลดจากเซิร์ฟเวอร์ของ Hugging Face)

### 2. กำหนดโทเคนพิเศษและขนาดของ embedding

โทเคน pad\_token ถูกกำหนดให้เป็น eos\_token (End of Sequence) เพื่อใช้เติมคำในกรณีที่ข้อมูลมีความยาวไม่เท่ากันในแต่ละตัวอย่าง (batch) และกำหนดขนาดของ embedding ในโมเดลให้ตรงกับจำนวนโทเคนใน tokenizer โดยใช้ resize\_token\_embeddings

### 3. บันทึกข้อมูลลงไฟล์ชั่วคราว

ในส่วนนี้ ข้อมูลที่ได้รับ (data) จะถูกเขียนลงในไฟล์ข้อความ (temp\_train\_data.txt) โดยแยกข้อมูลแต่ละรายการด้วยบรรทัดใหม่ (\n)

### 4. สร้าง Dataset

บรรทัดนี้สร้าง TextDataset ซึ่งใช้สำหรับการจัดการชุดข้อมูล โดยกำหนดให้ tokenizer ใช้ในการแปลงข้อความเป็นโทเคน และใช้ block\_size=128 ซึ่งหมายถึงแต่ละบล็อกของข้อมูลจะมีความยาว 128 โทเคน

### 5. กำหนดตัวช่วยในการประมวลผลข้อมูล (Data Collator)

ใช้ DataCollatorForLanguageModeling เพื่อจัดการการสร้าง batch ของข้อมูลที่จะใช้ในการเทรน โดยกำหนด mlm=False (ไม่ใช่ Masked Language Modeling) เพราะ GPT-2 ไม่ใช่โมเดลที่ใช้เทคนิคนี้ (MLM ใช้ในโมเดลอย่าง BERT)

### 6. กำหนดพารามิเตอร์การเทรน (TrainingArguments)

- output\_dir คือโฟลเดอร์ที่จะบันทึกผลลัพธ์จากการเทรน
- overwrite\_output\_dir=True หมายถึงจะเขียนทับผลลัพธ์เดิมในโฟลเดอร์นี้ทุกครั้ง
- num\_train\_epochs=4 จะเทรนทั้งหมด 4 รอบ
- per\_device\_train\_batch\_size=2 คือขนาด batch ของข้อมูลต่ออุปกรณ์ (เช่นต่อ GPU) ตั้งเป็น 2
- gradient\_accumulation\_steps=4 ใช้สะสม gradients 4 ขั้นตอนก่อนที่จะอัปเดตโมเดล
- save\_steps=500 จะบันทึกสถานะโมเดลทุกๆ 500 steps (จริงๆ โมเดลที่เทรนแต่ละครั้งไม่ถึง 500 จึงเป็นเซฟทั้งหมด)

- `save_total_limit=2` จะเก็บสถานะโมเดลล่าสุดเพียง 2 ชุด
- `eval_strategy="steps"` ระบุว่าประเมินผลในระหว่างการเทรนทุกๆ 500 steps
- `load_best_model_at_end=True` จะโหลดโมเดลที่ดีที่สุดเมื่อจบการเทรน
- `logging_steps=1` จะบันทึกค่า log ทุกๆ 1 step

#### 7. กำหนด Optimizer

ใช้ AdamW เป็นตัวปรับแต่งโมเดล (optimizer) โดยใช้ learning rate เท่ากับ  $5e-5$

#### 8. กำหนด Callback

มีการใช้ EarlyStoppingCallback เพื่อตรวจจับว่าเมื่อไรที่การเทรนควรหยุดก่อนจะครบ epochs หากประสิทธิภาพไม่ดีขึ้น โดยมีความอดทน (patience) อยู่ที่ 3 epochs นอกจากนี้ยังใช้ PlottingCallback เพื่อสร้างกราฟการเรียนรู้

#### 9. สร้าง Trainer และเริ่มการเทรน

Trainer เป็นคลาสที่จัดการขั้นตอนการเทรนทั้งหมด โดยรับพารามิเตอร์ต่างๆ ที่กำหนดไว้ข้างต้น ได้แก่ โมเดล, พารามิเตอร์การเทรน, ตัวจัดการข้อมูล, optimizer และ callbacks. การเรียก `trainer.train()` จะเริ่มกระบวนการเทรนโมเดล

สรุปวิธีสร้างโมเดล

1. เตรียมชุดข้อมูลใหม่ (ในที่นี้คือ dataset)
2. โหลดโมเดล GPT-2 ที่ถูกฝึกมาก่อนแล้ว
3. ใช้ Trainer จาก Hugging Face เพื่อเทรนโมเดลโดยใช้ชุดข้อมูลใหม่
4. ปรับแต่งพารามิเตอร์ต่างๆ เช่น optimizer – learning rate, ขนาด batch, จำนวน epochs และ Gradient accumulation

## 7. การประเมินโมเดล

```
def evaluate_model(model, tokenizer, test_data, dataset):
    generated_stories = []
    bleu_scores = []

    for i, words in enumerate(test_data, 1):
        story = generate_story(model, tokenizer, words)
        generated_stories.append(story)

        matching_row = dataset[
            (dataset['word1'] == words[0]) &
            (dataset['word2'] == words[1]) &
            (dataset['word3'] == words[2]) &
            (dataset['word4'] == words[3]) &
            (dataset['word5'] == words[4])
        ]

        print(f"\nTest {i}:")
        print(f"Input words: {' '.join(words)}")
        print(f"Generated story: {story}")

        if not matching_row.empty:
            reference_story = matching_row['story'].values[0]
            bleu_score = calculate_bleu(reference_story, story)
            bleu_scores.append(bleu_score)
            print(f"Reference story: {reference_story}")
            print(f"BLEU score: {bleu_score:.4f}")
        else:
            print("Warning: No matching story found in the dataset.")
            bleu_scores.append(None)

    print("-" * 50)

    valid_bleu_scores = [score for score in bleu_scores if score is not None]
    average_bleu = np.mean(valid_bleu_scores) if valid_bleu_scores else 0
    print(f"\nAverage BLEU Score: {average_bleu:.4f}")
    print(f"Total test cases: {len(test_data)}")
    print(f"Valid BLEU scores: {len(valid_bleu_scores)}")

    return generated_stories, bleu_scores
```

ส่วนนี้ใช้ในการประเมินผลการทำงานของโมเดล โดยการนำเรื่องราวที่โมเดลสร้างขึ้นมาเปรียบเทียบกับเรื่องราวจริงใน dataset พร้อมทั้งคำนวณค่า BLEU score เพื่อวัดความใกล้เคียง

## 8. การทดสอบโมเดลกับคำที่ป้อนโดยผู้ใช

```
def check_words_in_story(words, story):
    missing_words = [word for word in words if word.lower() not in story.lower()]
    return missing_words

user_input = input("Please enter 5 words: ")
words = user_input.split()
if len(words) != 5:
    print("Please enter exactly 5 words.")
missing_words = words.copy()

attempts = 0
max_attempts = 5

while len(missing_words) > 2 and attempts < max_attempts:
    story = generate_story(model, tokenizer, words)
    missing_words = check_words_in_story(words, story)

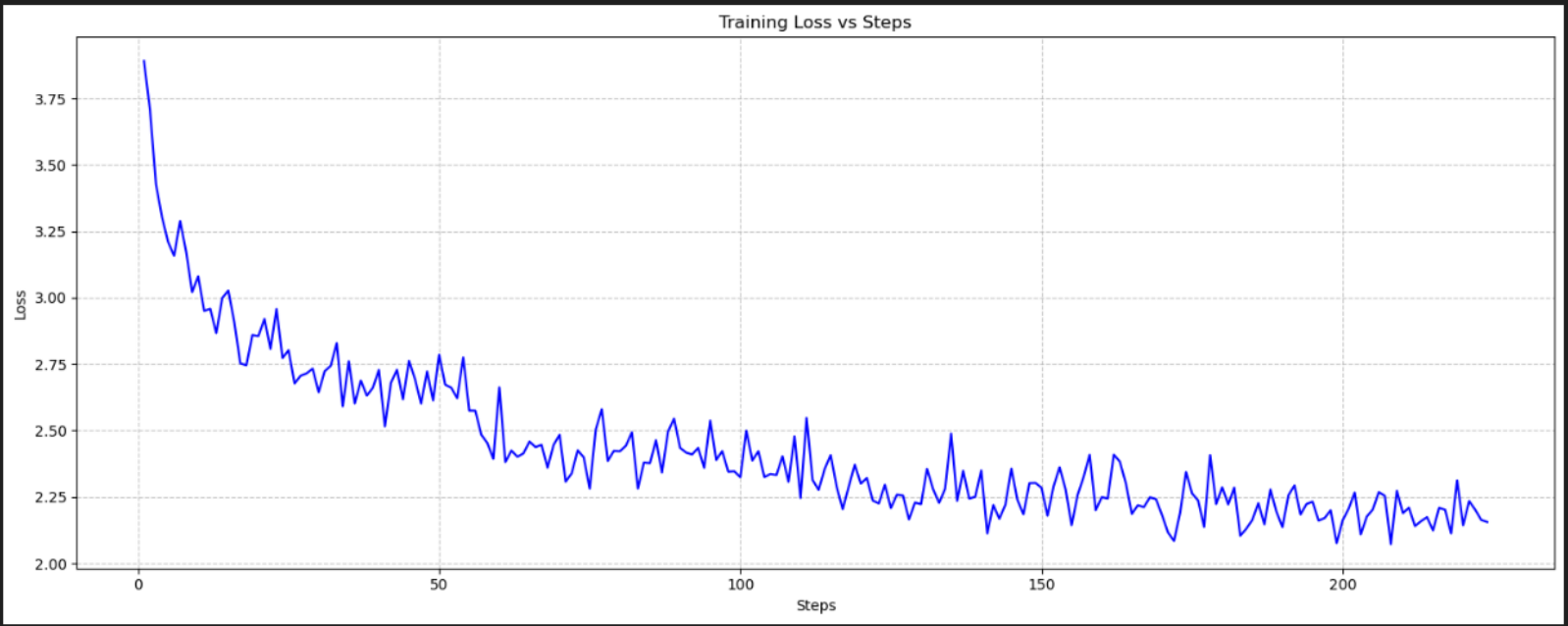
    if len(missing_words) <= 2:
        print("\nInput words:", ' '.join(words))
        print("\nGenerated story:")
        print(story)
        if missing_words:
            print(f"\nNote: The word(s) '{', '.join(missing_words)}' are not in the story.")
            break
    else:
        print(f"\nAttempt {attempts + 1}: Missing words: {' ', '.join(missing_words)}")
        print("Generating a new story...")
        attempts += 1

if attempts == max_attempts:
    print("\nFailed to generate a story with 3 or more words after multiple attempts.")
    print("Last generated story:")
    print(story)
    print(f"Missing words: {' ', '.join(missing_words)}")
```

ฟังก์ชันนี้จะสร้างเรื่องราวจากคำ 5 คำที่ผู้ใช้ป้อน และตรวจสอบว่ามีคำไหนบ้างที่ไม่ปรากฏในเรื่องราว โดยจะถ้า generate แล้วไม่พบคำที่ป้อนเข้าไปเกิน 2 คำ จะทำการ generate ใหม่ และจะทำแบบนี้ไปเรื่อยๆ 5 ครั้ง ถ้าไม่ได้จริงๆ จะนำผลลัพธ์สุดท้ายมาแสดงแทน

## ตัวอย่างตอนเทรน

```
-----  
{'loss': 3.4255, 'grad_norm': 8.552093505859375, 'learning_rate': 4.933035714285715e-05, 'epoch': 0.05}  
Step: 4  
Loss: 3.3027  
-----  
{'loss': 3.3027, 'grad_norm': 8.183371543884277, 'learning_rate': 4.910714285714286e-05, 'epoch': 0.07}  
Step: 5  
Loss: 3.21  
-----  
{'loss': 3.21, 'grad_norm': 7.396580696105957, 'learning_rate': 4.888392857142857e-05, 'epoch': 0.09}  
Step: 6  
Loss: 3.1578  
-----  
{'loss': 3.1578, 'grad_norm': 7.60649299621582, 'learning_rate': 4.866071428571429e-05, 'epoch': 0.11}  
Step: 7  
...  
Loss: None  
-----  
{'train_runtime': 1467.5074, 'train_samples_per_second': 1.224, 'train_steps_per_second': 0.153, 'train_loss': 2.4330369319234575, 'epoch': 3.98}  
Model and tokenizer saved to ./fine\_tuned\_model  
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```



## ตัวอย่างการทดลอง BLEU score

```
Test 157:
Input words: member really society spend tend
Generated story: The society needs to balance their values and interests, but they need to balance the benefits and risks involved. By focusing on the importance of fostering a fa
word1: include word2: in word3: job word4: business word5: in story: In the job market, the industry is known for its innovative solutions. As the industry expanded,
Reference story: Being a member of society means we really need to understand our roles. We tend to spend our time debating what makes a good citizen, yet often forget that our ac
BLEU score: 0.0000
-----
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `attention_mask` to obtain reliable results.
Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Test 158:
Input words: stay amount smile sign performance
Generated story: The audience was excited about the performance of the show and the impact they had on the company. The performances were memorable, demonstrating the value of the
word1: within word2: just word3: family word4: same word5: drive story: Within a family of seven, a drive to drive was just as
Reference story: To stay positive, we must focus on the amount of joy we can bring to others. A simple smile can be a sign of hope in a world that often overlooks the beauty in sm
BLEU score: 0.0000
-----
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `attention_mask` to obtain reliable results.
Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Test 159:
Input words: continue understand arm approach early
Generated story: The arm approach allowed the group to engage in more effective strategies. Their strategy emphasized that they would keep their focus on the goals they had set. A
word1: act word2: that word3: every word4: spend word5: find story: The act of spending money on personal items was a source of stress for many. Every step in
Reference story: To continue growing, we must understand the importance of an open arm approach, welcoming new ideas early on. The willingness to embrace change is what drives pro
BLEU score: 0.0000
-----
The attention mask and the pad token id were not set. As a consequence, you may observe unexpected behavior. Please pass your input's `attention_mask` to obtain reliable results.
Setting `pad_token_id` to `eos_token_id`:None for open-end generation.

Test 160:
Input words: attention character painting box range
Generated story: The character behind the character art was a popular character in the art market. The artist captured the moment with passion and enthusiasm, prompting the audien
word1: dog word2: many word3: in word4: the word
Reference story: In art, attention to detail can transform a simple painting into a powerful narrative. Each character within it tells a story that goes beyond the confines of a b
BLEU score: 0.0000
-----

Test 161:
Input words: child relationship tax leader in
Generated story: As the leader of the family business, she launched a successful campaign to raise awareness about child-protection issues. They sought to support families by offe
Reference story: Every child deserves a nurturing environment that fosters healthy relationships. A leader in this community plays a crucial role in shaping policies, from educati
BLEU score: 0.0000
-----

Average BLEU Score: 0.0036
Total test cases: 161
Valid BLEU scores: 161
Save 100% Results !
```

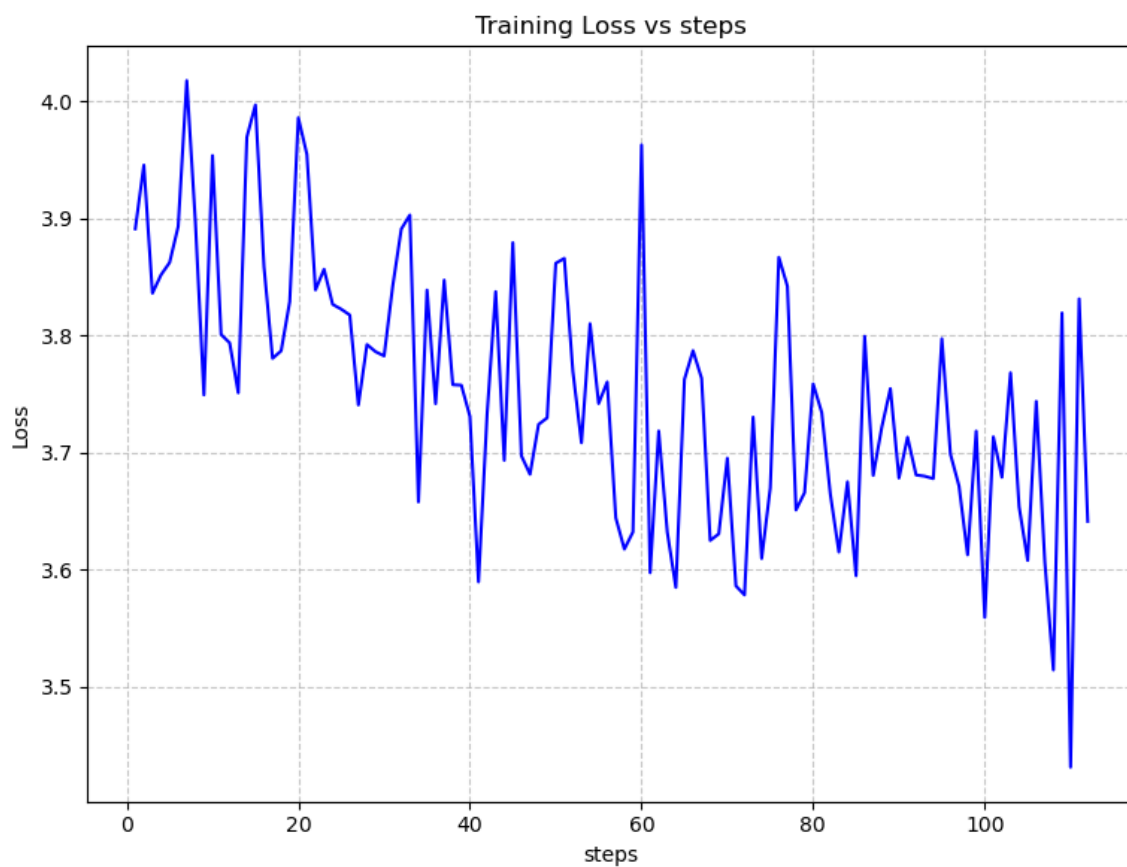
ตัวอย่างผลลัพธ์ BLEU score แบบไฟล์ .csv

together	word	shake	bill	language	The bill was a key piece of legislation that would strengthen community institutions. It would allow for education and outreach, ensuring that everyone could benefit from the fruits of the local economy. Together, they would craft a new model for education. Together they would create a platform that fostered creativity and resilience. word1: stand word2: in word3: now word4: trial word5: stay story: Now, the trial began. The new	0.049245998618317355
room	store	position	large	staff	In the room, it was hard not to notice a small, loyal customer waiting for a cup. The owner sat down and observed the importance of loyalty in their business. She was committed to maintaining high standards and keeping customers satisfied. Employees shared stories, showing how loyal customers felt. It was a testament to how loyal they were to their business and the community. word1: car word2: in word3: many word4: just word	4.728137398348123e-155
sort	improve	short	only	join	In the bustling city of New York, a woman named Sallie, a sort of middle-aged woman, began to search for a new career. Her dreams of becoming a professional were dashed, however, as she remained focused on her journey, focusing on her dreams. As she faced obstacles, she learned to navigate her way forward, guiding others toward success. word1: buy word2: save word3: like word4: be	3.954266712463121e-155
increase	hot	benefit	city	bit	In a world of globalization, the benefits of travel came first. Travel was crucial for many countries, but for those who wanted to stay longer, it was essential. The capital city was bustling with tourists, bustling with food and entertainment. The bustling streets were bustling with art and art, and the warmth of the capital hung in the air. word1: break word2: to word3: every word4: in word5: new story	4.803946774143399e-155
enough	lay	pretty	above	now	The story of the new neighborhood was inspiring, with residents sharing stories about how it transformed their neighborhood. The development envisioned a new community center, with a new art gallery and a new plaza. The community felt empowered to create their own projects, and they were ready to take the next step. word1: more word2: in word3: need word4: for word5: future story: In a bustling city, a new industry was	5.027979501587917e-155
almost	simple	responsibility	commercial	during	After a small presentation, the group decided to present a simple idea that would make the media laugh. They had to explain the importance of responsibility in the world, especially in the media landscape. They brainstormed ideas that would unite audiences and create awareness. The audience was excited, eager to hear their ideas. word1: many word2: more word3: word4: many story: The many voices expressing their opinions often led to controversies.	0.04731054312719297
less	realize	month	eye	light	less realized month day word1: word2: from word3: to word4: present word5: at story: from story: the event was about to occur at the end of the month, and it was a rare opportunity to present a new idea. They were inspired by the beautiful surroundings, the vibrant colors of the countryside, and the chance to experience a new perspective. The result was a celebration of art and culture, showcasing some	0.06639671070426985
painting	easy	amount	picture	set	The artist took a picture of a painting and decided to go with the simple approach. He wanted to create a picture that felt more natural than the typical. He chose a simple line, focusing on detail and beauty. He created a beautiful painting, with a simple, yet beautiful, connection to the viewer. word1: age word2: change word3: from word4: not word5: next story: The age of change in the	4.9419994486183515e-155

## การทดลองที่ 1 : ทดลองปรับ Learning rate

โดยการทดลองนี้เราจะมีกำหนดค่าของ hyperparameter ดังนี้

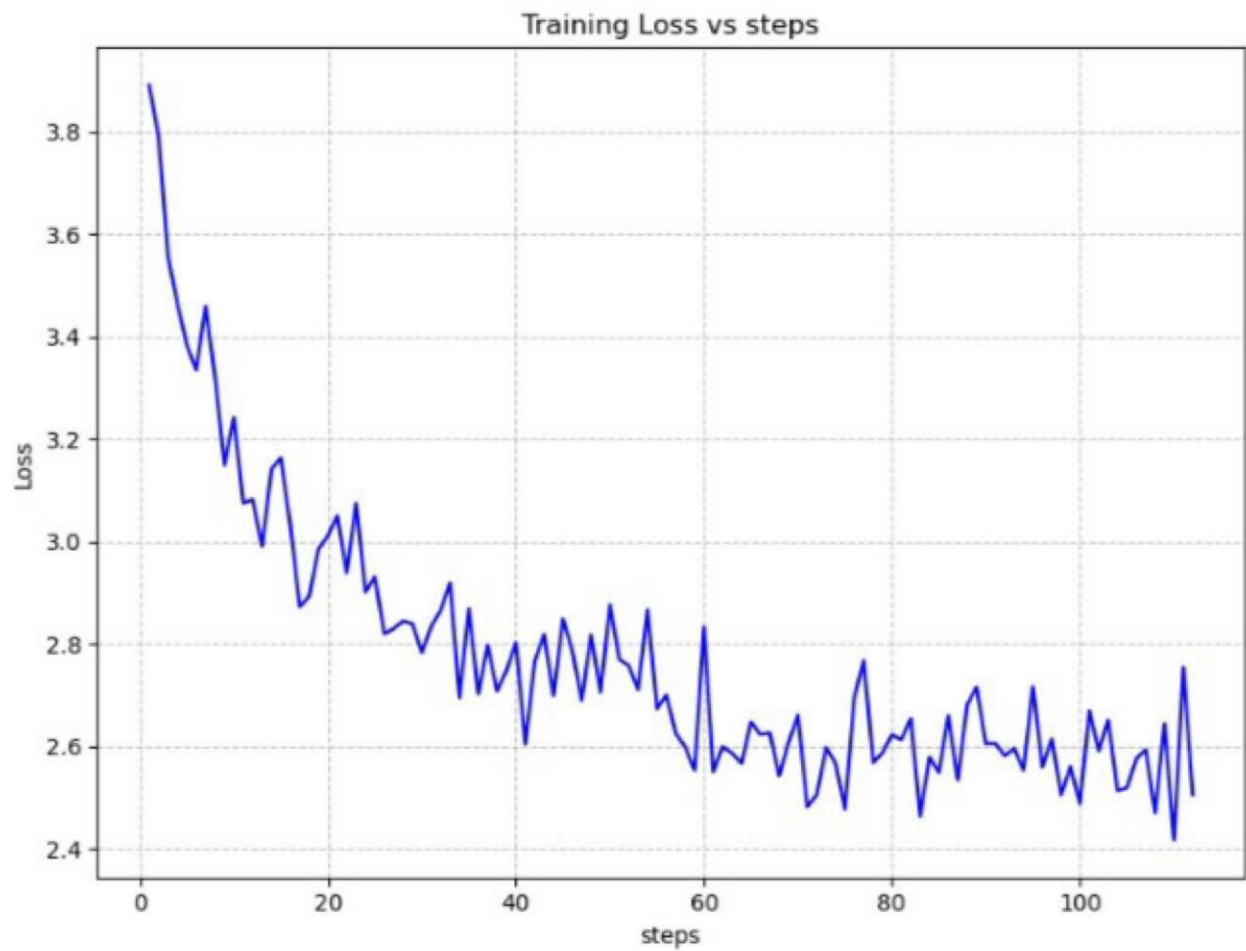
- Learning rate =  $x$  (ตามการทดลอง)
- Batch size = 2
- Gradient accumulation = 4
- Epoch = 2



กราฟที่ 1.1 เมื่อปรับค่า Learning rate =  $1e-6$

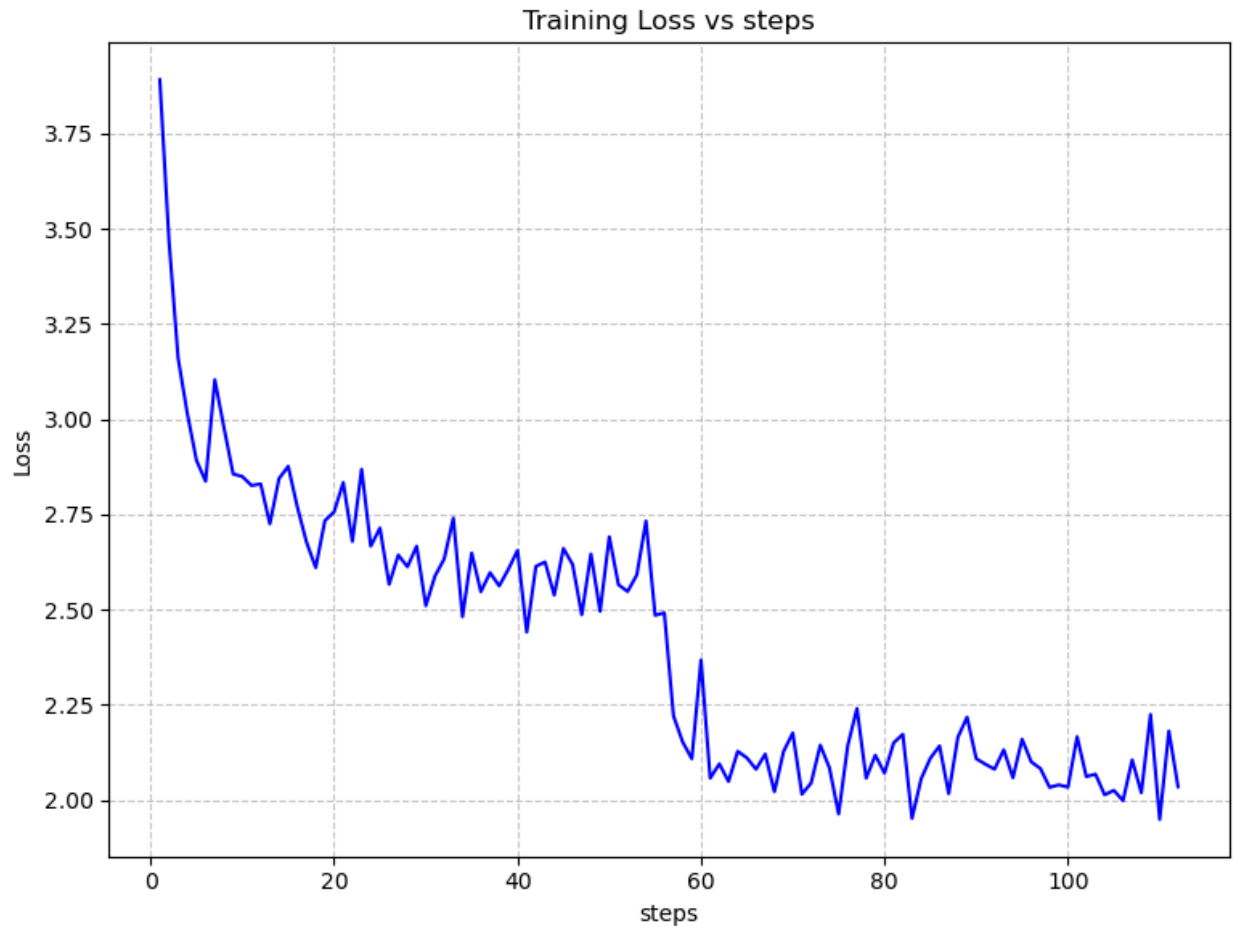
เมื่อ Learning rate มีค่าเท่ากับ  $1e-6$  จะสังเกตเห็นว่ากราฟมีความผันผวนมาก และยังมีค่า Training Loss ที่เทียบกับ Steps ลดลงน้อยมาก จึงทำให้เห็นว่า ค่า Learning rate =  $1e-6$  มีค่าที่น้อยเกินไป





กราฟที่ 1.2 เมื่อปรับค่า Learning rate =  $1e-5$

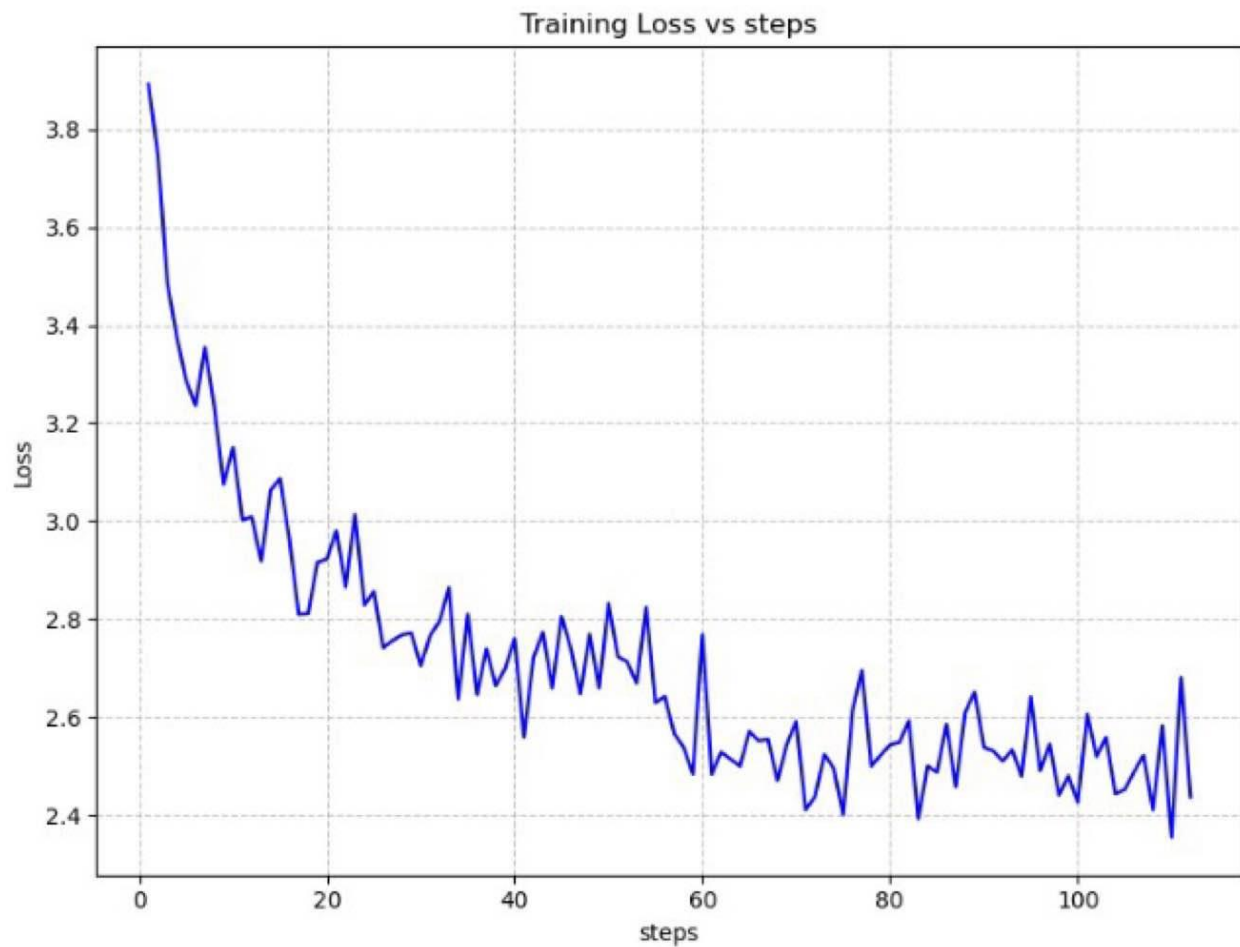
เมื่อ Learning rate มีค่าเท่ากับ  $1e-5$  จะสังเกตเห็นว่ากราฟมีความผันผวนเล็กน้อย และยังมีค่า Training Loss ที่เทียบต่อ Steps ลดลงในระดับที่พอรับได้ จึงทำให้เห็นว่า ค่า Learning rate =  $1e-5$  มีค่าที่พอรับได้



กราฟที่ 1.3 เมื่อปรับค่า Learning rate =  $1e-4$

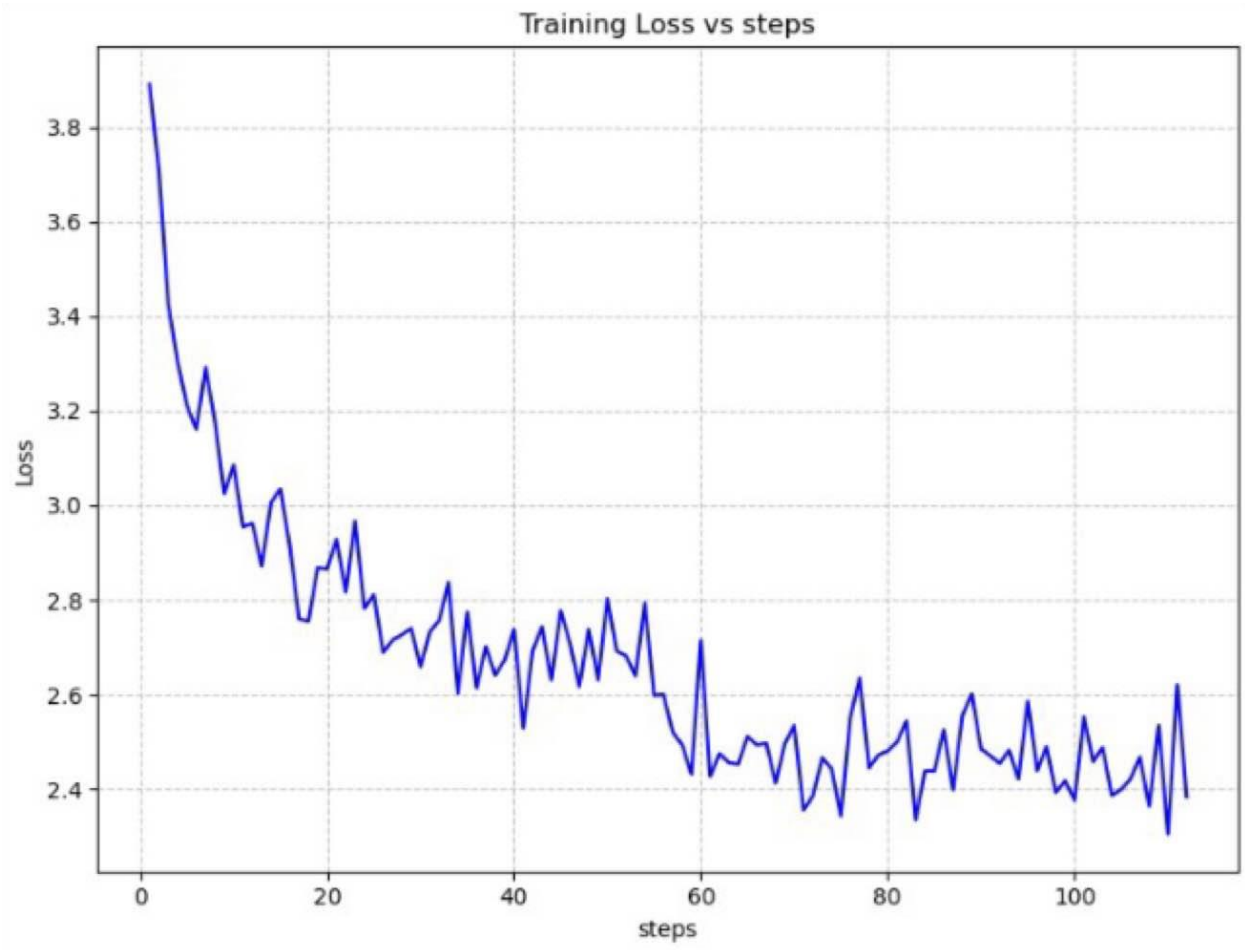
เมื่อ Learning rate มีค่าเท่ากับ  $1e-4$  จะสังเกตเห็นว่ากราฟมีความผันผวนบางช่วงที่สูงในช่วง steps = 55 - 60 และยังมีค่า Training Loss ที่เทียบต่อ Steps ลดลงในระดับที่สูง แต่เมื่อพิจารณาจากกราฟ กราฟมีความผันผวนที่สูง ซึ่งอาจไม่เป็นผลดีในระยะยาว จึงทำให้ทราบได้ว่า ค่า Learning rate =  $1e-5$  มีค่าที่สูงเกินไป

หลังจากที่เราได้รู้ค่า Learning rate =  $1e-5$  มีค่าที่สูงเกินไป กับ Learning rate =  $1e-6$  มีค่าที่น้อยเกินไป และค่า Learning rate =  $1e-5$  มีค่าที่พอรับได้ จึงทำให้ทราบว่า ค่า Learning rate ที่อยู่ในระดับพอดี คือ  $1e-5$  เพราะฉะนั้นเราจะทำการทดลองที่ ค่า  $2e-5$  -  $9e-5$  โดยจะหาช่วงที่เป็นไปได้และนำมาทดลอง



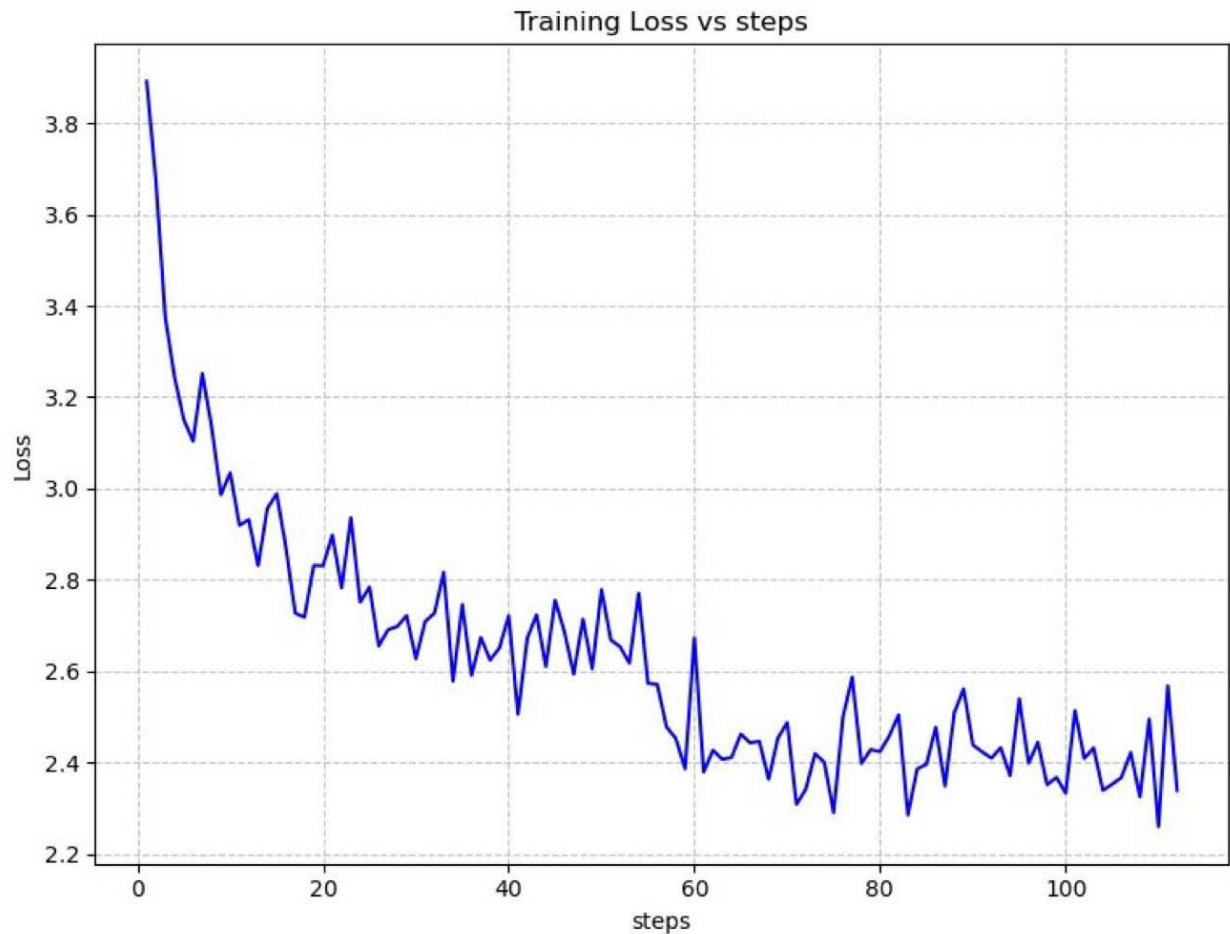
กราฟที่ 1.4 เมื่อปรับค่า Learning rate =  $4e-5$

เมื่อ Learning rate มีค่าเท่ากับ  $4e-5$  จะสังเกตเห็นว่ากราฟมีความผันผวนเล็กน้อย เหมือนกราฟที่มี Learning rate =  $1e-5$  และยังมีค่า Training Loss ที่เทียบกับ Steps ลดลงในระดับที่พอรับได้และลดลงได้มากกว่า Learning rate =  $1e-5$  อยู่เล็กน้อย จึงทำให้เห็นว่า ค่า Learning rate =  $4e-5$  มีค่าที่พอรับได้ และควรเลือก Learning rate ที่มีค่าเท่ากับ  $4e-5$  มากกว่า Learning rate ที่มีค่าเท่ากับ  $1e-5$



กราฟที่ 1.5 เมื่อปรับค่า Learning rate =  $5e-5$

เมื่อ Learning rate มีค่าเท่ากับ  $5e-5$  จะสังเกตเห็นว่ากราฟมีความผันผวนเล็กน้อย เหมือนกราฟที่มี Learning rate =  $4e-5$  และยังมีค่า Training Loss ที่เทียบกับ Steps ลดลงในระดับที่พอรับได้และลดลงได้มากกว่า Learning rate =  $4e-5$  อยู่เล็กน้อย จึงทำให้เห็นว่า ค่า Learning rate =  $5e-5$  มีค่าที่พอรับได้ เหมือน Learning rate =  $4e-5$



กราฟที่ 1.6 เมื่อปรับค่า Learning rate =  $6e-5$

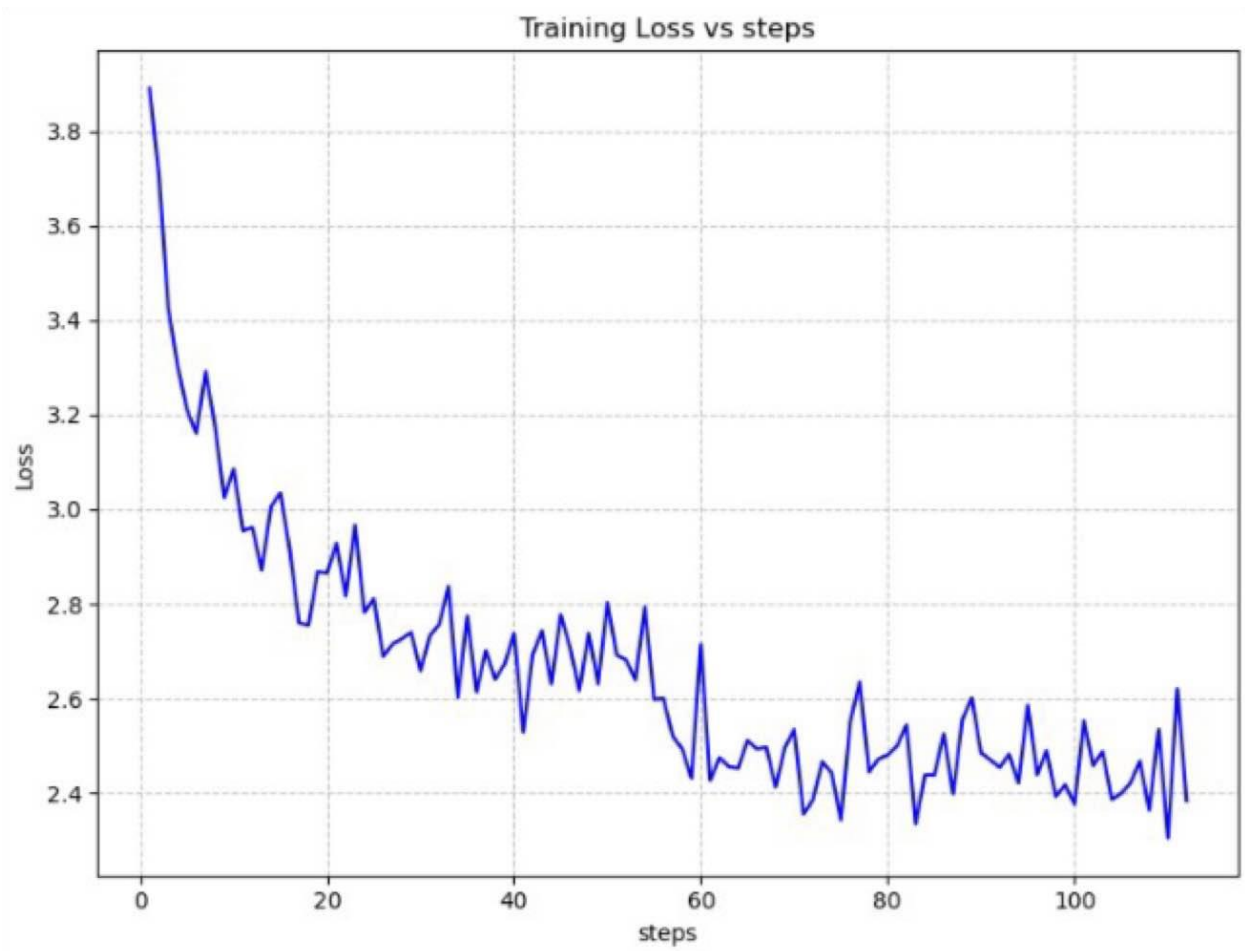
เมื่อ Learning rate มีค่าเท่ากับ  $6e-5$  จะสังเกตเห็นว่ากราฟมีความผันผวนเล็กน้อยแต่มากกว่ากราฟที่มี Learning rate =  $6e-5$  และยังมีค่า Training Loss ที่เทียบกับ Steps ลดลงในระดับที่พอรับได้และลดลงได้มากกว่า Learning rate =  $5e-5$  อยู่เล็กน้อย จึงทำให้เห็นว่า ค่า Learning rate =  $6e-5$  มีค่าที่พอรับได้

โดยจากการทดลองครั้งนี้ทำให้รู้ว่าคุณค่า Learning rate ในช่วง  $4e-5$  ถึง  $6e-5$  นั้น มีกราฟที่คล้ายคลึงกันมาก และมีการลดลงของ Training Loss ที่พอๆกัน แต่ทำให้เห็นว่ากราฟที่มีค่า Learning rate ที่สูงจะทำให้ค่า Training Loss ลดลงเร็วก็จริง แต่ทำให้กราฟมีความผันผวนมากขึ้น และเมื่อ Learning rate ที่ต่ำจะทำให้กราฟมีความผันผวนน้อย แต่ก็ต้องแลกกับ Training ที่ลดลงช้า เพราะฉะนั้นการทดลองนี้สรุปได้ว่า ค่า Learning rate =  $5e-5$  เป็นค่าที่ดีที่สุดสูงในการเทรนครั้งนี้

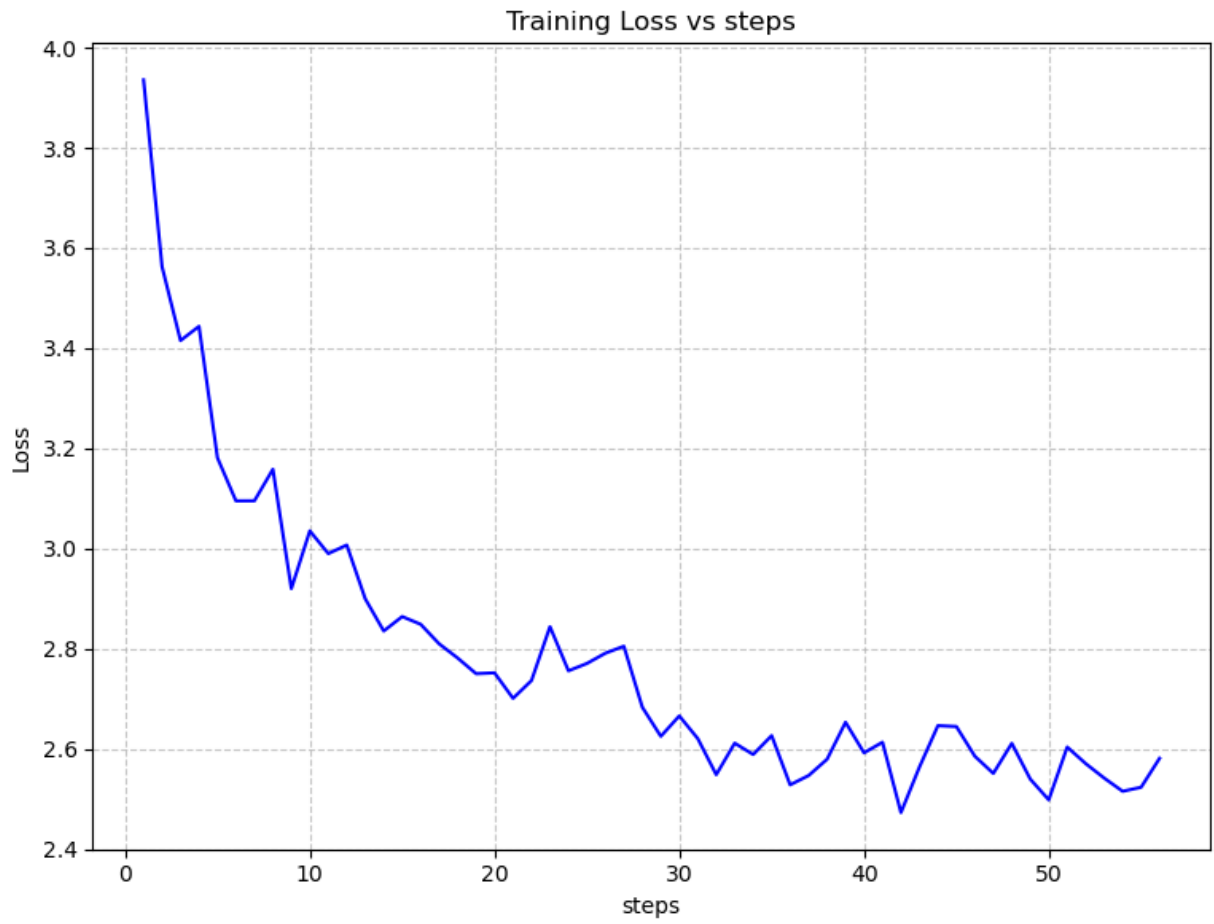
## การทดลองที่ 2 : ทดลองปรับ Batch size

โดยการทดลองนี้เราจะมีกำหนดค่าของ hyperparameter ดังนี้

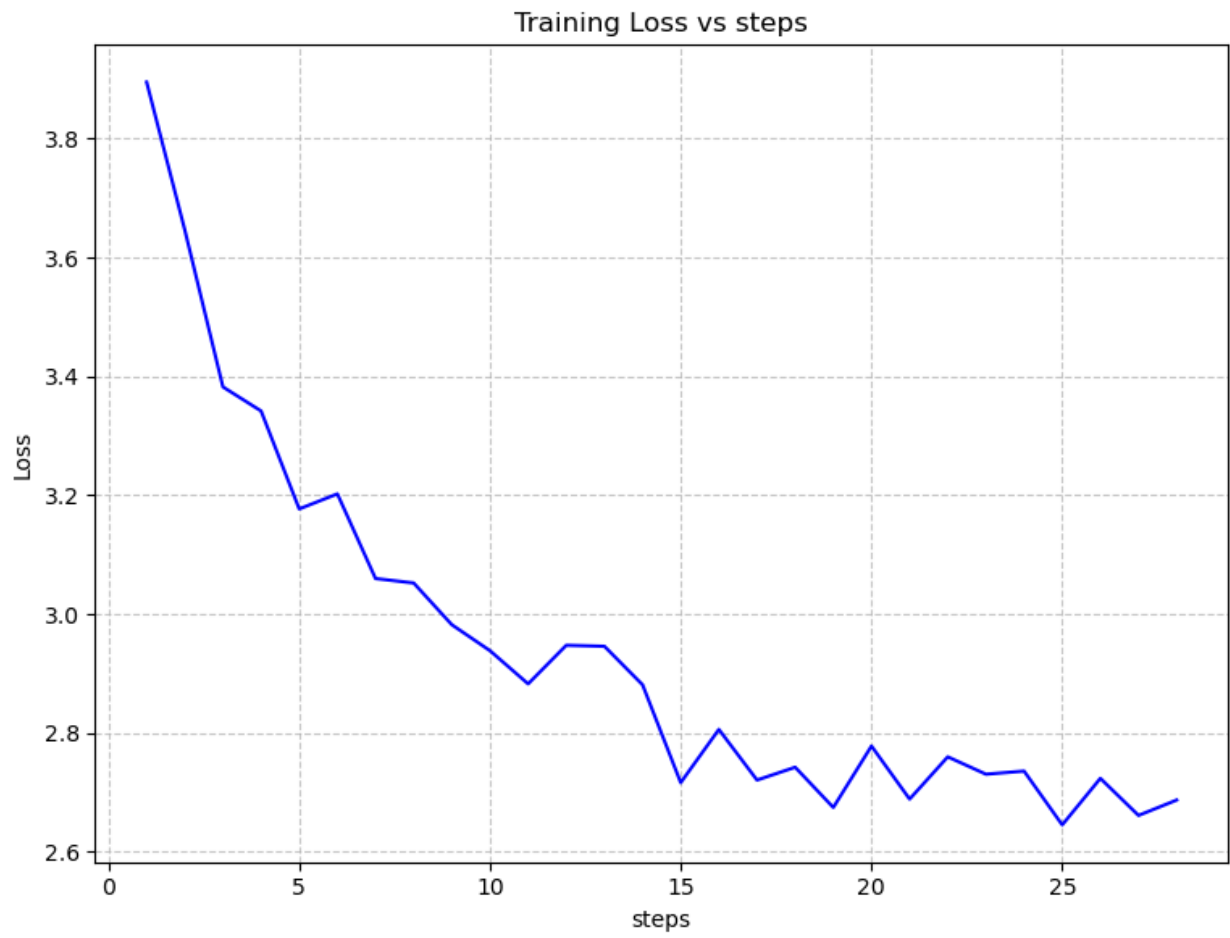
- Learning rate =  $5e-5$
- Batch size = x (ตามการทดลอง)
- Gradient accumulation = 4
- Epoch = 2



กราฟที่ 2.1 เมื่อปรับค่า Batch size = 2

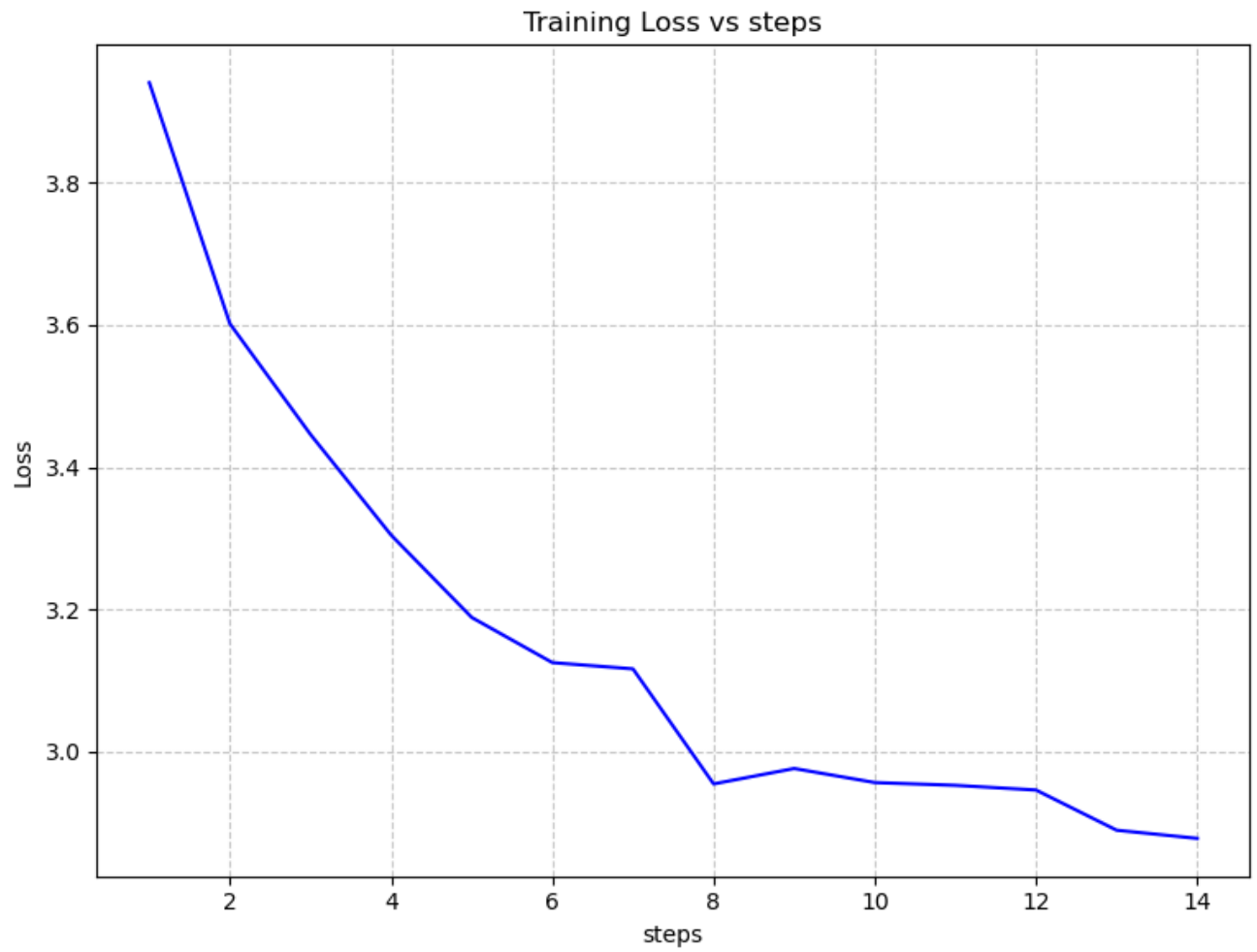


กราฟที่ 2.2 เมื่อปรับค่า Batch size = 4

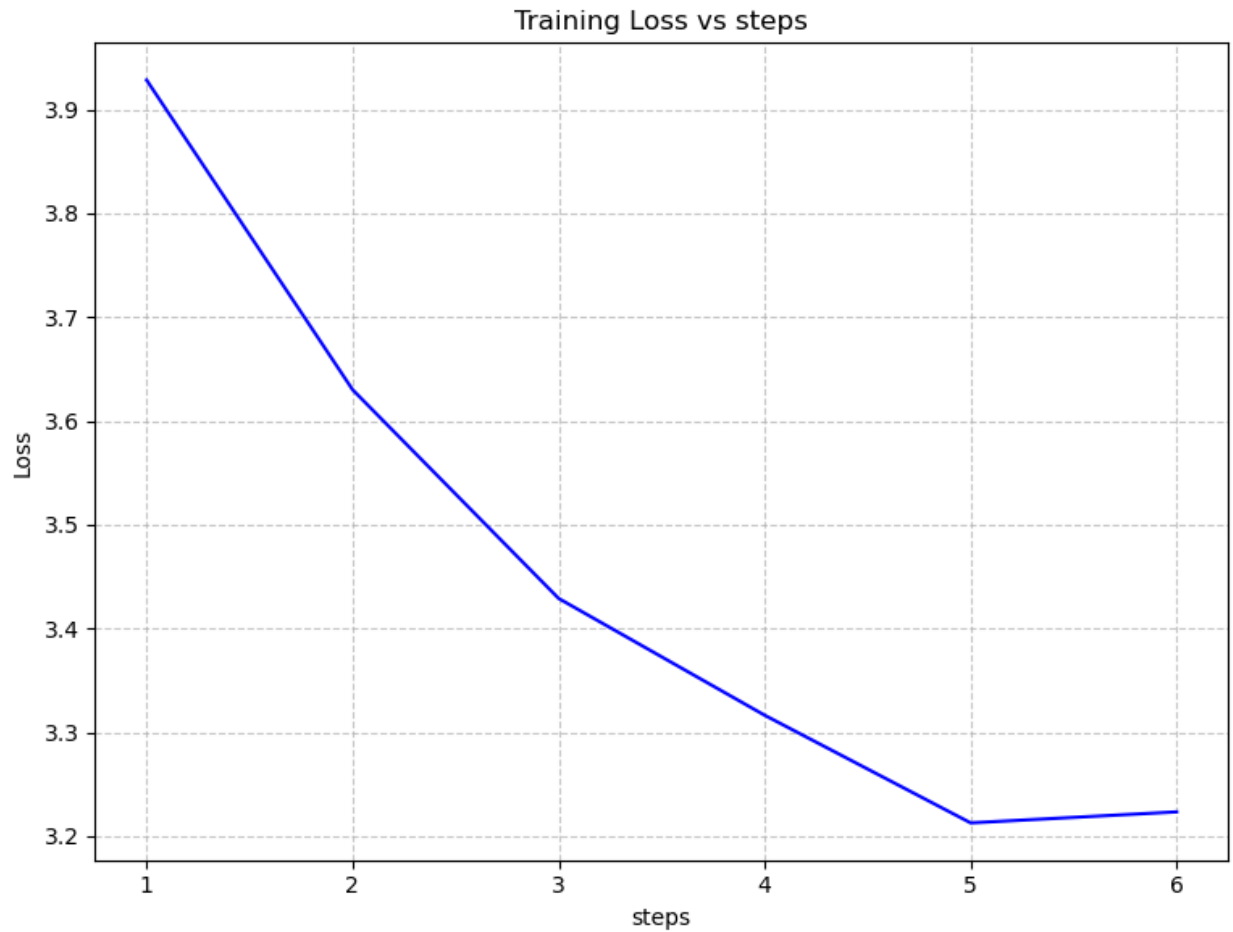


กราฟที่ 2.3 เมื่อปรับค่า Batch size = 8





กราฟที่ 2.4 เมื่อปรับค่า Batch size = 16



กราฟที่ 2.5 เมื่อปรับค่า Batch size = 32

จากการทดลองเมื่อดูกราฟทั้ง 5 กราฟที่กล่าวไปข้างต้นจะสังเกตเห็นว่า ความผันผวนของกราฟนั้น ลดลงอย่างเห็นได้ชัดเมื่อมีการเพิ่มค่า Batch size ที่ละ 2 เท่า แต่ค่า Training Loss นั้นก็ลดลงไปด้วยอย่างเห็นได้ชัด และยังคงใช้งาน CPU และ GPU มากขึ้นตามไปด้วย แต่ข้อดีคือ ใช้เวลาน้อยลง และจำนวน step น้อยลงตามไปด้วย ซึ่งกราฟก็ยังไม่ได้สรุปเกี่ยวกับคุณภาพของการเทรนในครั้งนี้ได้ โดยเราจะทำการทดลองใหม่โดยการเทียบจากค่า Bleu score โดยจะเปลี่ยนค่าของ epoch จาก 2 เป็น 3 เนื่องจาก epoch = 2 มีค่าของ Bleu score ที่ต่ำมากจนไม่สามารถเอามาเปรียบเทียบกันได้ ซึ่งสรุปได้ตามตารางด้านล่าง

โดยการทดลองนี้เราจะมีกำหนดค่าของ hyperparameter ดังนี้

- Learning rate =  $5e-5$
- Batch size = x (ตามการทดลอง)
- Gradient accumulation = 4
- Epoch = 3

Batch size	Average BLEU Score
2	0.0019
4	0.0014
8	0.0014
16	0.0010
32	0.0005

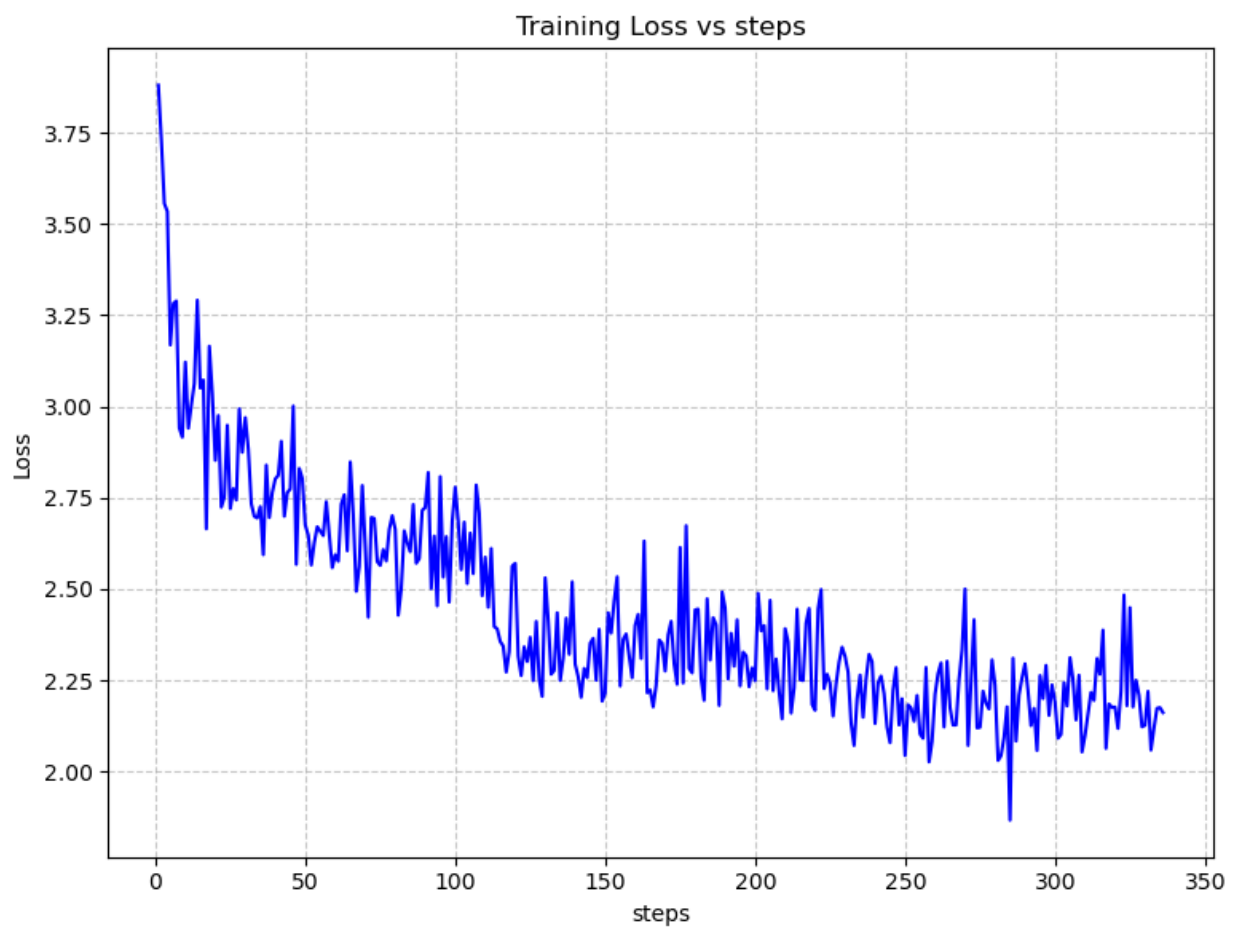
ตารางที่ 1 ค่า Average BLEU score จากการทดลอง Batch Size

จากตารางจะสังเกตเห็นว่าค่า Batch size ที่มี Average BLEU score มากที่สุดคือ Batch size = 2 เพราะฉะนั้นค่า Batch size ที่ดีที่สุดสำหรับการเทรนในครั้งนี้ คือ 2

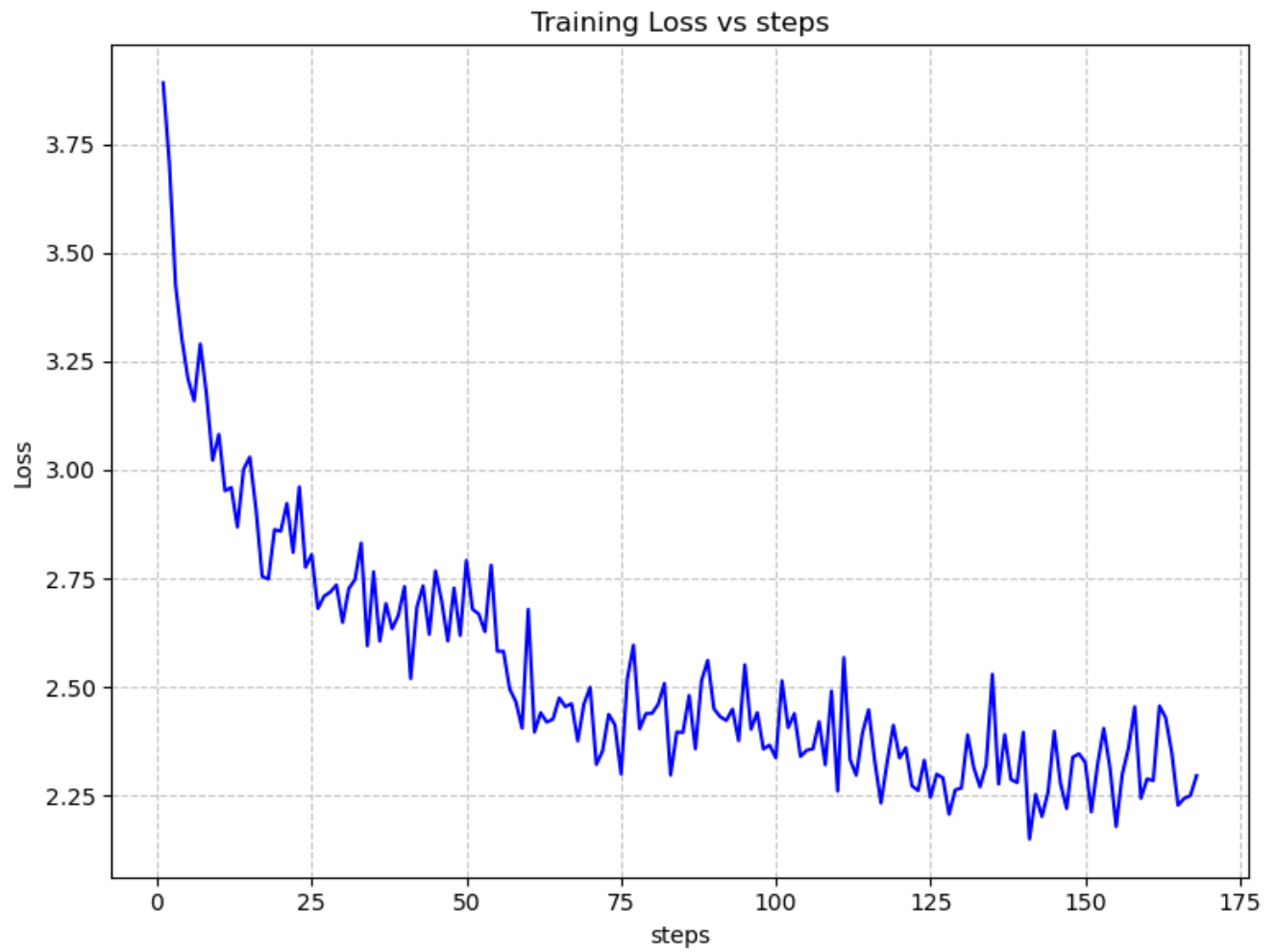
### การทดลองที่ 3 : ทดลองปรับ Gradient Accumulation

โดยการทดลองนี้เราจะมีกำหนดค่าของ hyperparameter ดังนี้

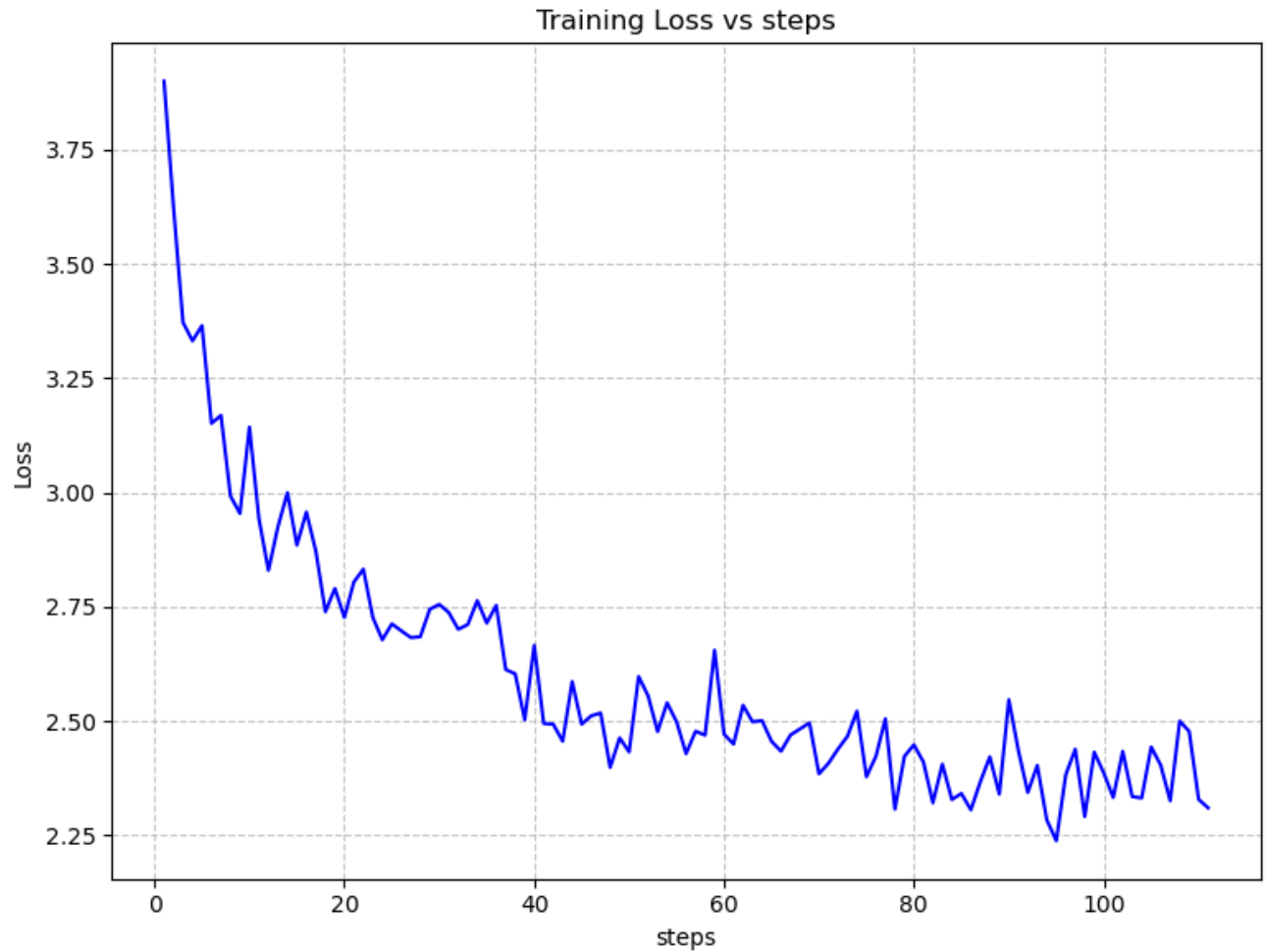
- Learning rate =  $5e-5$
- Batch size = 2
- Gradient accumulation = x (ตามการทดลอง)
- Epoch = 3



กราฟที่ 3.1 เมื่อปรับค่า Gradient accumulation = 2



กราฟที่ 3.2 เมื่อปรับค่า Gradient accumulation = 4



กราฟที่ 3.3 เมื่อปรับค่า Gradient accumulation = 6

จากกราฟก็ยังไม่สรุปไม่ได้ว่าควรเลือก Gradient accumulation เท่ากับเท่าไร เนื่องจากกราฟที่มี Gradient accumulation น้อยมีแนวโน้มที่จะลด Training loss ได้เยอะ แต่ก็มี step ที่มากขึ้น และความผันผวนของกราฟก็มากขึ้นตาม เพราะฉะนั้นจึงจำเป็นต้องเทียบจากการทดลองหาค่า Average BLEU score ผลดังตารางด้านล่าง

Gradient accumulation	Average BLEU Score
2	0.0009
4	0.0019
6	0.0006

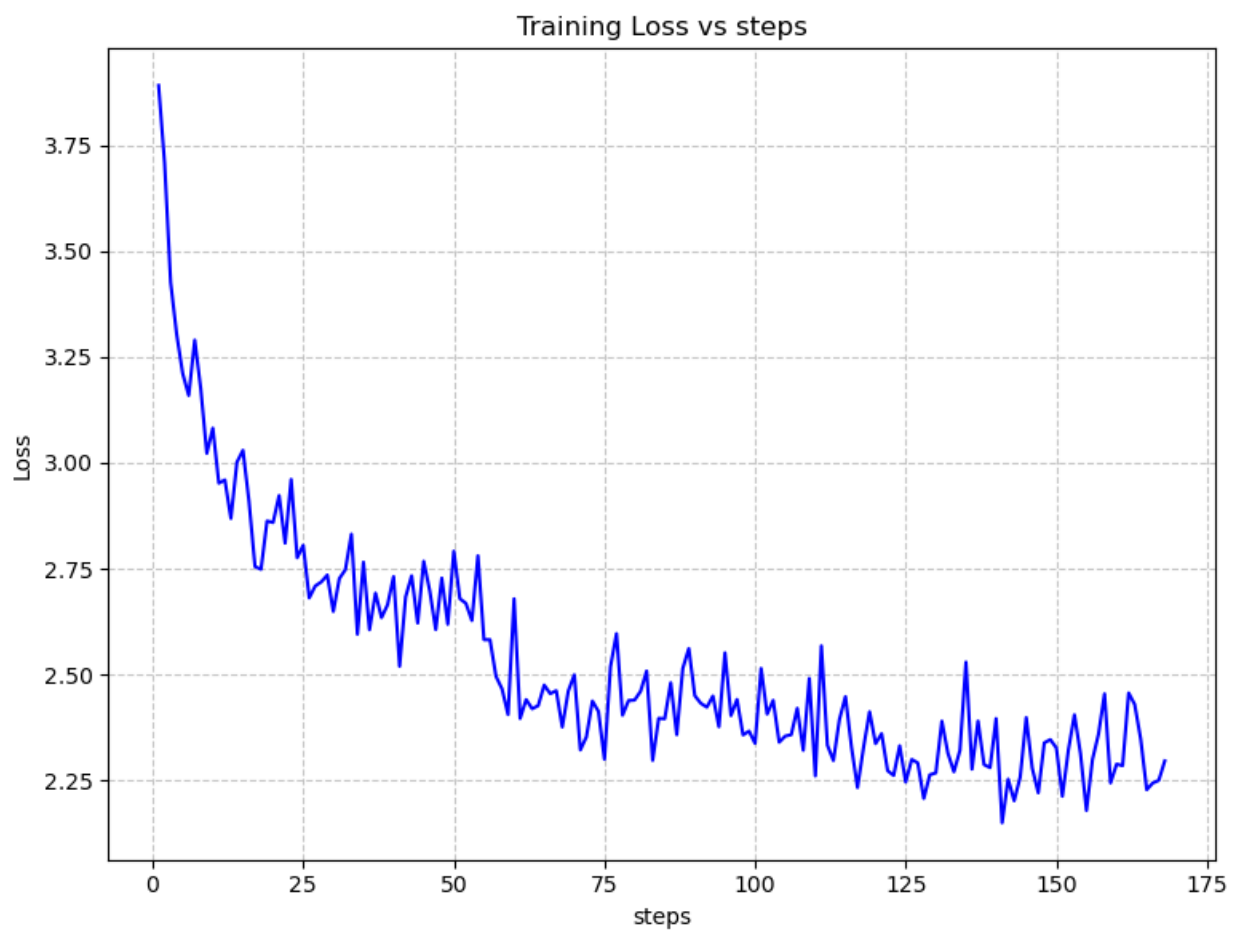
ตารางที่ 2 ค่า **Average BLEU score** จากการทดลองปรับค่า Gradient accumulation

จากตารางจะสังเกตเห็นว่าค่า Gradient accumulation ที่มี **Average BLEU score** มากที่สุดคือ Gradient accumulation = 4 เพราะฉะนั้นค่า Gradient accumulation ที่ดีที่สุดสำหรับการเทรนในครั้งนี้ คือ 4

#### การทดลองที่ 4 : ทดลองปรับ Epoch

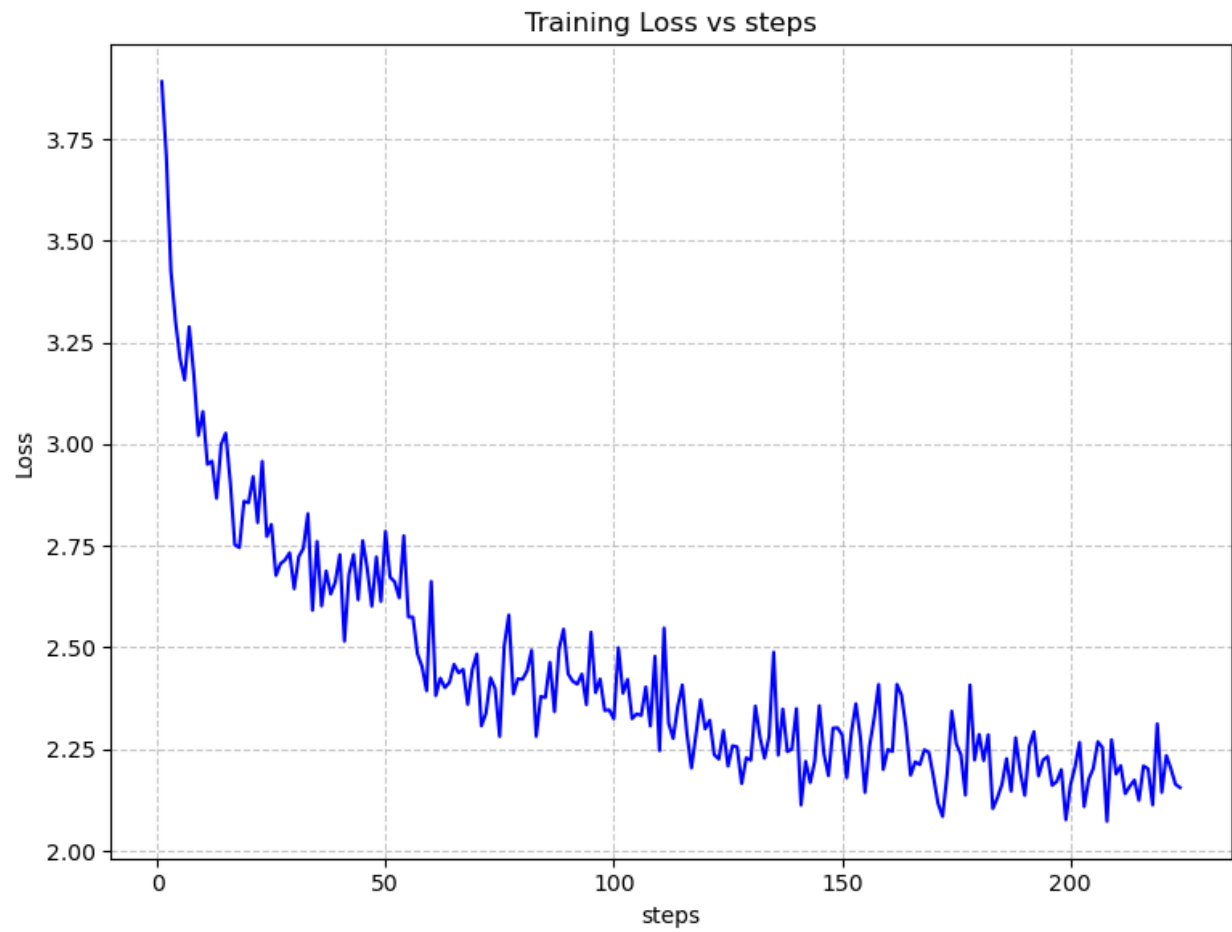
โดยการทดลองนี้เราจะมีกำหนดค่าของ hyperparameter ดังนี้

- Learning rate =  $5e-5$
- Batch size = 2
- Gradient accumulation = 4
- Epoch = x (ตามการทดลอง)

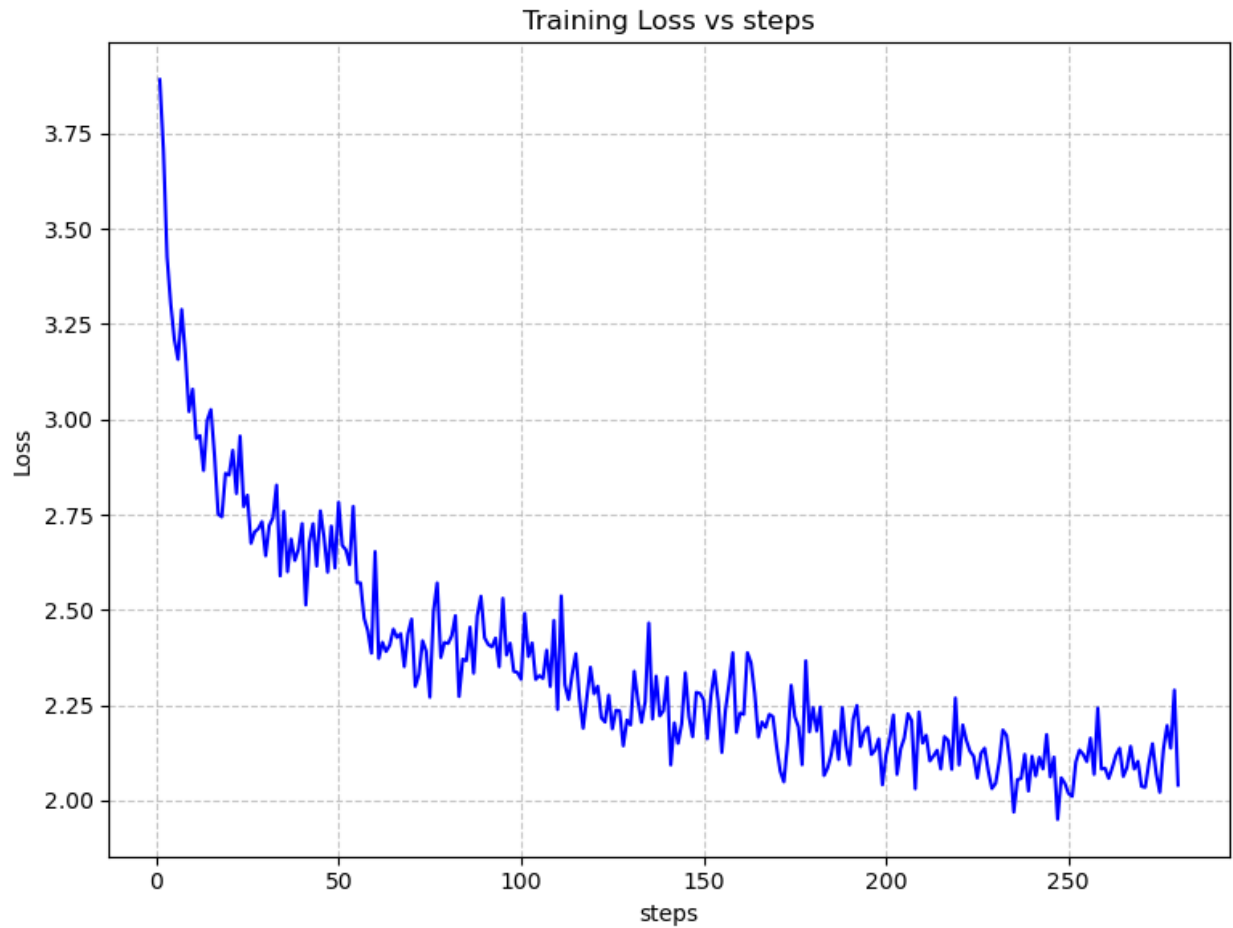


กราฟที่ 4.1 เมื่อปรับค่า Epoch = 3





กราฟที่ 4.2 เมื่อปรับค่า Epoch = 4



กราฟที่ 4.3 เมื่อปรับค่า Epoch = 5

จากกราฟทั้งหมดก็ยังสามารถสรุปได้ว่าจะใช้ epoch เท่ากับเท่าไร เพราะไม่ได้บ่งบอกถึงคุณภาพในการเทรน เพราะฉะนั้นต้องทดลองหา Average BLEU score ต่อ ดังตารางด้านล่าง

Epoch	Average BLEU Score
3	0.0019
4	0.0038
5	0.0015

ตารางที่ 3 ค่า Average BLEU score จากการทดลองปรับค่า Gradient accumulation

## สรุปผลการทดลอง

ค่า hyperparameter ที่ทำให้เทรนได้ดีที่สุดใน project ชุดนี้ คือ

- Learning rate =  $5e-5$
- Batch size = 2
- Gradient accumulation = 4
- Epoch = 4

โดยใช้ pre-trained model คือ gpt2

Optimizer คือ Adam

เทรนโดยใช้กระบวนการ Fine-tuning

ตัวอย่าง ผลลัพธ์ที่ได้ เมื่อป้อนคำ kill , hammer , human , face ,eye

```
Input words: kill hammer human face eye

Generated story:
The hammer was a reminder of the importance of human life and the importance in the world. The human eye would
word1: stop word2: word3: in word4: over word5: the story: In the end, the

Note: The word(s) 'kill' are not in the story.
```

Input words: kill hammer human face eye

Generated story:

The hammer was a reminder of the importance of human life and the importance in the world. The human eye would be wiped from the landscape, but it remained calm as they watched the fierce hammer fall. The head of the team prepared to face the challenges head on, but they knew they would face whatever came next.

word1: stop word2: word3: in word4: over word5: the story: In the end, the

Note: The word(s) 'kill' are not in the story.

โดยข้อผิดพลาดที่เห็นได้ชัด คือ เมื่อเรากำหนดความยาวของประโยค หรือ ตัวแปร max\_length

```
def generate_story(model, tokenizer, words):
    input_text = f"Words: {' '.join(words)}\nStory:"
    input_ids = tokenizer.encode(input_text, return_tensors='pt')

    output = model.generate(
        input_ids,
        max_length=100,
        num_return_sequences=1,
        no_repeat_ngram_size=3,
        top_k=50,
        top_p=0.95,
        temperature=0.7,
        do_sample=True
    )
    generated_text = tokenizer.decode(output[0], skip_special_tokens=True)
    # story = generated_text.split('Story:')[1].strip()
    story = generated_text.split('Story:')[1].strip().replace('<|endoftext|>', '').strip()

    return story
```

อาจเกิดข้อผิดพลาดดังกล่าวข้างต้น เนื่องจากไม่สามารถหาคำเชื่อมให้ ครบถึง 100 ได้เลยทำการ generate story ตาม trainset ที่ให้ ที่มีการกำหนดคำต่อในบรรทัดถัดไป และข้อผิดพลาดอย่างที่สอง คือ บางครั้งอาจจะไม่สามารถ generate story ได้ครบตามคำที่ได้มา เพราะ trainset มีเนื้อหาที่ครอบคลุมแค่บางส่วน หรือบางคำ

### ข้อเสนอแนะ

1. ควรมี dataset ให้มากขึ้น
2. Dataset ควรเป็นเนื้อหาที่ไม่หลากหลาย หรือ เฉพาะทาง
3. ควรมี GPU และ CPU ที่มีประสิทธิภาพสูงเนื่องจากการเทรน 1 ครั้ง ใช้เวลานานอย่างมาก
4. ความยาวของ Story แต่ละชุดใน dataset ควรมีความยาวที่เท่าๆกัน เพื่อให้ง่ายในการเทรน
5. ควรลองเปลี่ยน Pre-trained model มาเทรนดูเพื่อทดสอบประสิทธิภาพ แต่คอมพิวเตอร์ของผู้จัดทำ ไม่ได้มีประสิทธิภาพสูงขนาดที่ เวลาในการเทรนต่ำกว่า 2 ชม ได้ในโมเดล Pre-trained model GPT2 ที่ ขนาดใหญ่ขึ้นไป (GPT2-medium , GPT2-Large , GPT2-XL)

### ขอบคุณข้อมูลที่นำมาประกอบ Dataset

- [fourmilab.ch/etexts/www/warworlds/](http://fourmilab.ch/etexts/www/warworlds/)
- [blog.reedsy.com/short-story](http://blog.reedsy.com/short-story)
- [hthepatronsaintsofsuperheroes.wordpress.com/](http://hthepatronsaintsofsuperheroes.wordpress.com/)
- [ncbi.nlm.nih.gov/](http://ncbi.nlm.nih.gov/)
- [learnenglishkids.britishcouncil.org](http://learnenglishkids.britishcouncil.org)
- [americanliterature.com](http://americanliterature.com)

### ขอบคุณข้อมูลที่นำมาประกอบการศึกษา

- [linkedin.com/pulse/creating-text-generation-model-using-gpt-2-anurag-dhiman-yuvbc](https://linkedin.com/pulse/creating-text-generation-model-using-gpt-2-anurag-dhiman-yuvbc)

สอนการทำ text generate

- [stackoverflow.com/questions/37101114/what-to-download-in-order-to-make-nltk-tokenize-word-tokenize-work](https://stackoverflow.com/questions/37101114/what-to-download-in-order-to-make-nltk-tokenize-word-tokenize-work)

สอนการทำ text generate วิธีใช้ nltk หรือ Natural Language Toolkit

- [discuss.huggingface.co/t/logs-of-training-and-validation-loss/1974](https://discuss.huggingface.co/t/logs-of-training-and-validation-loss/1974)

สอนการปรับ hyperparameter

- [huggingface.co/docs/transformers/en/main\\_classes/callback](https://huggingface.co/docs/transformers/en/main_classes/callback)

สอนการปรับ hyperparameter

- [stackoverflow.com/questions/78302444/calculating-bleu-score-between-candidate-and-reference-sentences-in-python](https://stackoverflow.com/questions/78302444/calculating-bleu-score-between-candidate-and-reference-sentences-in-python)

สอนการหา BLEU Score

- chatgpt , cluade

ช่วยหาจุดบัก และข้อแก้ไขนำไปสู่การค้นหาข้อมูล หรือ library อื่นๆมาใช้