

Fine-Tuning OpenAI API model for specific task

Final Project – Coding in AI 2023
King Mongkut's University of Technology Thonburi

Introduction



- Fine-tuning OpenAI API model สำหรับงานเฉพาะเจาะจง
- โดเมนเฉพาะเจาะจง คือการ ให้ข้อมูลเกี่ยวกับสถานที่ท่องเที่ยวใน
จังหวัดเชียงใหม่ จากนั้นจะนำชุดข้อมูลของสถานที่ท่องเที่ยวใน
จังหวัดเชียงใหม่ มาทำการ fine-tuning ให้กับแต่ละ โมเดลของ OpenAI

ที่มารูปภาพ: https://cdnlogo.com/logo/openai_120588.html

<https://thai.tourismthailand.org/Attraction/ม่อนแจ่ม>



Theory and Techniques

- Unsupervised Fine-tuning
- Pre-training Model
- Data Collection
- Fine-tuning pre-trained model



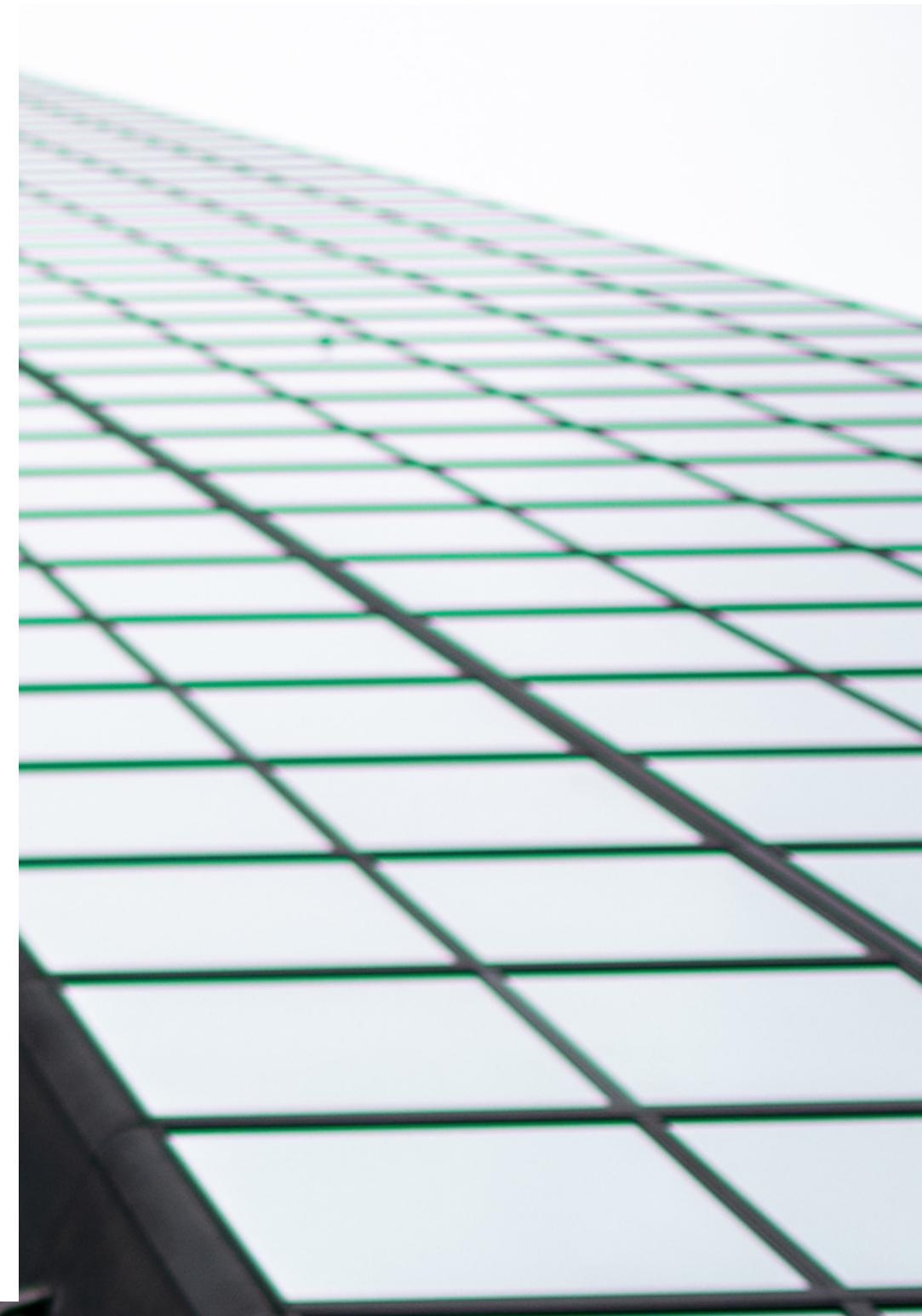
ที่มารูปภาพ: <https://www.tourismthailand.org/Destinations/Provinces/chiang-mai/101>

Unsupervised Fine-tuning

- Unsupervised Fine-tuning จะช่วยให้โมเดลสามารถปรับให้เข้ากับโดเมนหรือข้อกำหนดเฉพาะโดยไม่มี label เนพาะงานที่ชัดเจน

Pre-training Model

- การ Pre-training เป็นพื้นฐานของความสามารถของโมเดล OpenAI API ช่วยให้เข้าใจภาษา และบรรยายในรูปแบบ dataset ที่หลากหลาย สำหรับประสิทธิผลของการ fine-tune
- โมเดลที่เลือกมาใช้ใน OpenAI API: GPT-3.5-Turbo-1106 และ Babbage-002





GPT-3.5-Turbo-1106

- เป็นโมเดลที่ใหญ่และมีประสิทธิภาพมากด้วยพารามิเตอร์มากกว่า 175 พันล้านพารามิเตอร์ ต้องใช้ทรัพยากรการประมวลผลในระดับสูงมากเพื่อการ train และทำนาย
- สำหรับองค์กรส่วนใหญ่แล้ว GPT-3.5 Turbo ถือเป็น overkill สำหรับการใช้งานทั่วไปในส่วนโมเดลขนาดเล็กกว่าอย่าง Babbage และ Davinci จึงคุ้นค่ามากกว่า

Babbage-002

- เป็นโมเดลปัญญาประดิษฐ์สำหรับการสนทนา มีพารามิเตอร์จำนวน 1.3 พันล้านตัว ทำให้เป็นโมเดลขนาดกลาง
- Babbage ถูกออกแบบมาสำหรับการใช้งานจริง เช่น chatbot และ assistant ใช้ข้อมูลและการประมวลผลน้อยกว่าโมเดลอื่นหลายเท่าในการฝึก

Data Collection

เทคนิค Web-Scraping เป็นวิธีการที่ใช้ในการดึงข้อมูลจากเว็บไซต์เพื่อรวบรวมข้อมูลเฉพาะงานเพื่อปรับแต่งโมเดล OpenAI API อย่างละเอียด

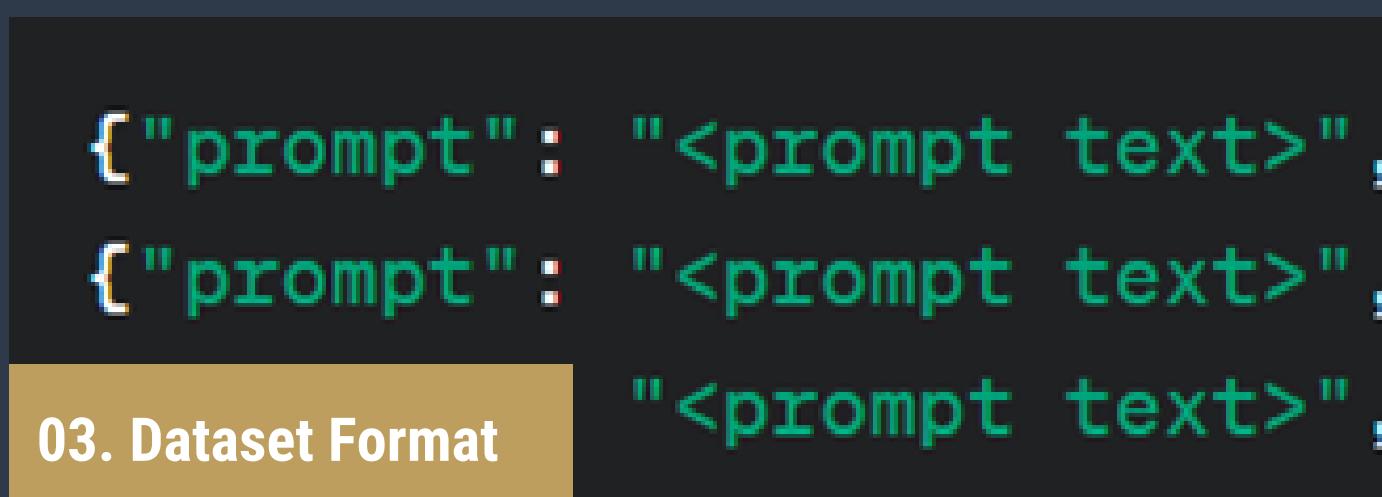
1. ข้อมูลที่เราเลือกมาใช้จะมาจากเว็บไซต์ท่องเที่ยวของการท่องเที่ยวแห่งประเทศไทย Tourism Authority of Thailand (TAT) ซึ่งเป็นหน่วยงานที่รับผิดชอบในการสนับสนุนและส่งเสริมกิจกรรมท่องเที่ยวในประเทศไทย
2. เนื่องจากแหล่งข้อมูลที่ใช้งานเป็นเว็บไซต์แบบโควตาไมค์เว็บ กระบวนการนี้ เราจึงเลือกใช้ไลบรารี Selenium จากนั้นเราจะไปสร้าง dataset จากข้อมูลที่ดึงมาจากเว็บเพื่อทำการ Fine-Tune OpenAI API model
3. รูปแบบ dataset ควรสอดคล้องกับข้อกำหนดอินพุตของโมเดล โดย dataset format จะอยู่ในรูปแบบ JSON Lines



The logo for 'amazing THAILAND' features the word 'amazing' in pink at the top, followed by 'THAILAND' in large, stylized letters where each letter has a yellow and pink gradient swoosh underneath it. Below the logo is a small gold bar containing the text '01. Data Sources'.



The Selenium logo consists of a green square icon with a white checkmark and the word 'Selenium' in green. Below the logo is a gold bar containing the text '02. Selenium Library'.



A dark background with three lines of JSON code in light blue. The first line starts with '{ "prompt": "<prompt text>"'. The second line starts with '{ "prompt": "<prompt text>"'. The third line starts with '<prompt text>'.

Below the JSON lines is a gold bar containing the text '03. Dataset Format'.

```
{"messages": [  
    {"role": "system", "content": system_content[i]},  
    {"role": "user", "content": user_content[i]},  
    {"role": "assistant", "content": assistant_content[i]}  
]
```

รูปแบบ dataset ที่ใช้สำหรับโมเดลเวอร์ชั่นใหม่นี้

- "messages": Key นี้จะแสดง list message ที่มีการแยกเปลี่ยนกันในการสนทนา แต่ละ message จะมี "role" และ "content"
- "role": Key นี้จะระบุ role ของผู้ส่งข้อความ ไม่ว่าจะเป็น "system", "user" หรือ "assistant"
- "content": Key นี้ประกอบด้วย content หรือ text ที่เป็นข้อเท็จจริงของ message

```
{"prompt": user_content[i], "completion": assistant_content[i]}
```

รูปแบบ dataset ที่ใช้สำหรับโมเดลเวอร์ชั่นเก่า

- "prompt": Key นี้แสดงถึงอินพุตที่ให้กับโมเดลภาษา เป็นข้อความหรือบริบทที่ต้องการให้โมเดลสร้างการตอบกลับ โดย prompt เป็นจุดเริ่มสำหรับโมเดลในการสร้างหรือเติมข้อความ
- "completion": Key นี้แสดงถึงเอกสารที่คาดหวังที่ต้องการให้โมเดลสร้างตาม prompt ที่กำหนด โดยท่านนี้ที่เป็นข้อมูลอ้างอิงหรือการ fine-tuning โดยการฝึกโมเดลในลักษณะที่เมื่อได้รับ prompt ที่คล้ายกัน โมเดลจะสร้างการตอบสนองใกล้เคียงกับข้อมูลที่ให้ไว้มากขึ้น

Fine-tuning Pre-trained Model

- 01 Prepare data**
- 02 Upload training and validation data**
- 03 Fine-tuning the model**
- 04 Use fine-tuned model**

1

Prepare data

แปลง dataset ให้เป็นรูปแบบ JSONL format ที่สามารถใช้กับการ Fine-tuning ได้

```
{"messages": [
    {"role": "system", "content": system_content[i]},
    {"role": "user", "content": user_content[i]},
    {"role": "assistant", "content": assistant_content[i]}
]}
```

2

Upload training and validation data

```
training_data = openai.files.create(
    file=open("training_data.jsonl", "rb"),
    purpose="fine-tune"
)

validation_data = openai.files.create(
    file=open("validation_data.jsonl", "rb"),
    purpose="fine-tune"
)
```

- อัปโหลดไฟล์ที่เป็น format ที่ตรงกับ API เข้าไปเก็บใน Base URL เพื่อใช้ในการนำไป fine-tuning โดยเราจะทำการแบ่ง training data กับ validation data ก่อนในกรณีที่ข้อมูลยังไม่ถูกแบ่ง

```
FileObject(id='file-c06ZMnMyGygwbWNFCedz3hi8',
FileObject(id='file-EsII02TMtG7ZQhTB3AtmCpct',
```

- เมื่ออัปโหลด data เสร็จแล้วก็จะสามารถดูไฟล์ໄວเดียวเพื่อนำมาใช้ในการ fine-tuning

3

Fine-tuning the model

```
model1 = openai.fine_tuning.jobs.create(  
    training_file="file-lUVKIZYrqECiJs23499L5a7T",  
    validation_file="file-LM6x5QnLCerJ846j53K1Y6VM",  
    model="gpt-3.5-turbo-1106",  
    suffix="gpt3.5turbomodel1",  
    hyperparameters={  
        "n_epochs": 3,  
        "batch_size": 8,  
        "learning_rate_multiplier": 0.01  
    }  
)
```

- ในการ fine-tuning ด้วยการใช้ API นี้ เราต้องทำการนำไฟล์ไอเดียส์ในพารามิเตอร์ รวมไปถึงการเลือกโมเดลการกำหนดชื่อโมเดลและการเลือก hyperparameters โดยการ fine-tuning OpenAI API model นี้ hyperparameters ที่ทำการปรับได้เมื่อเพียง epoch, batch size และ learning rate เท่านั้น

```
Data: {'step': 17, 'train_loss': 0.8631436824798584, 'valid_loss': 0.892247677842776, 'train_mean_token_accuracy': 0.7700554728507996, 'valid_mean_token_accuracy': 0.3517361111111114}  
Data: {'step': 18, 'train_loss': 0.8458332419395447, 'valid_loss': 0.8604461227433157, 'train_mean_token_accuracy': 0.7753578424453735, 'valid_mean_token_accuracy': 0.3538135593220339}  
Data: {'step': 19, 'train_loss': 0.8363702893257141, 'valid_loss': 0.7992794129357894, 'train_mean_token_accuracy': 0.7747975587844849, 'valid_mean_token_accuracy': 0.35288335517693314}  
Data: {'step': 20, 'train_loss': 0.858513355255127, 'valid_loss': 0.7559931666427261, 'train_mean_token_accuracy': 0.767275869846344, 'valid_mean_token_accuracy': 0.35549525101763907}
```

- เมื่อ fine-tuning เสร็จแล้วเราสามารถ monitor ค่า loss ในแต่ละ step ออกมานดูด้วยตัวเองได้ด้วยการใส่ fine-tuning job id ของโมเดลนี้ๆ ซึ่งค่าที่ถูก response ออกมายังอยู่ในพารามิเตอร์ data ซึ่งบอกค่า loss และค่า accuracy ของทั้ง training และ validation

4

Use fine-tuned model

```
1 import openai

1 # Set your OpenAI API key
2 api_key = "sk-jDrqUuXc1VMdQxbL7V8T3BlbkFJAt4rxXVc7mW8HkcFDfCj"
3
4 # Initialize the OpenAI client with your API key
5 openai.api_key = api_key
6
```

- นำ API key ของมาใส่เพื่อให้กาง Base URL สามารถรับ request และส่ง response อย่างถูกต้อง

```
1 response = openai.chat.completions.create(
2     model="ft:gpt-3.5-turbo-1106:personal:gpt3-5turbomodel1:8MY0VlsQ",
3     temperature=0,
4     max_tokens=1000,
5     messages=[
6         {"role": "system", "content": "You are a friendly and polite guide who give the travel place information in Thailand."},
7         {"role": "user", "content": "ชื่อสถานที่: ม่อนแจ่ม\ก\กประภท สถานที่ท่องเที่ยวโดยและภูเขา\ก\กข้อมูลรายละเอียดสถานที่: "}
8     ]
9 )
10 print(response.choices[0].message)
```

ChatCompletionMessage(content='สวัสดีครับ/ค่ะ! สถานที่ท่องเที่ยว "ม่อนแจ่ม" เป็นสถานที่ท่องเที่ยวที่ตั้งอยู่บนเทือกเขาดอยอ่างขางในจังหวัดเชียงใหม่ ม่อนแจ่มเป็นจุดชมวิวที่สวยงามที่สุดของดอยอ่างขาง')

- ในการนำมาใช้ generate text เพื่อดู output เราจะใช้คำสั่ง `openai.chat.completions.create()` ในการ request โดยจะใช้ชื่อบองโมเดลจาก Base URL มาใส่ในพารามิเตอร์ `model` เพื่อเลือกว่า จะใช้โมเดลไหน

Experimental Design and Results



- ทดสอบ Fine-tuning Dataset ที่แตกต่างกัน
- ทดสอบ Fine-tuning โมเดลที่ Hyperparameters แตกต่างกัน
- ทดสอบ Fine-tuning Pre-trained Model ที่แตกต่างกัน
- ทดสอบ Fine-tuning Dataset คุณภาษา

1

nodejs Fine-tuning Dataset ที่แตกต่างกัน

โมเดลที่ใช้ : gpt-3.5-turbo-1106

```
model1 = openai.fine_tuning.jobs.create(  
    training_file="file-lUVKIZYrqECiJs23499L5a7T",  
    validation_file="file-LM6xSQnLCerJ846jS3K1Y6VM",  
    model="gpt-3.5-turbo-1106",  
    suffix="gpt3.5turbomodel1",  
    hyperparameters={  
        "n_epochs": 3,  
        "batch_size": 8,  
        "learning_rate_multiplier": 0.01  
    }  
)
```

Dataset แบบแรกมีจำนวน data อยู่ที่ 60 samples

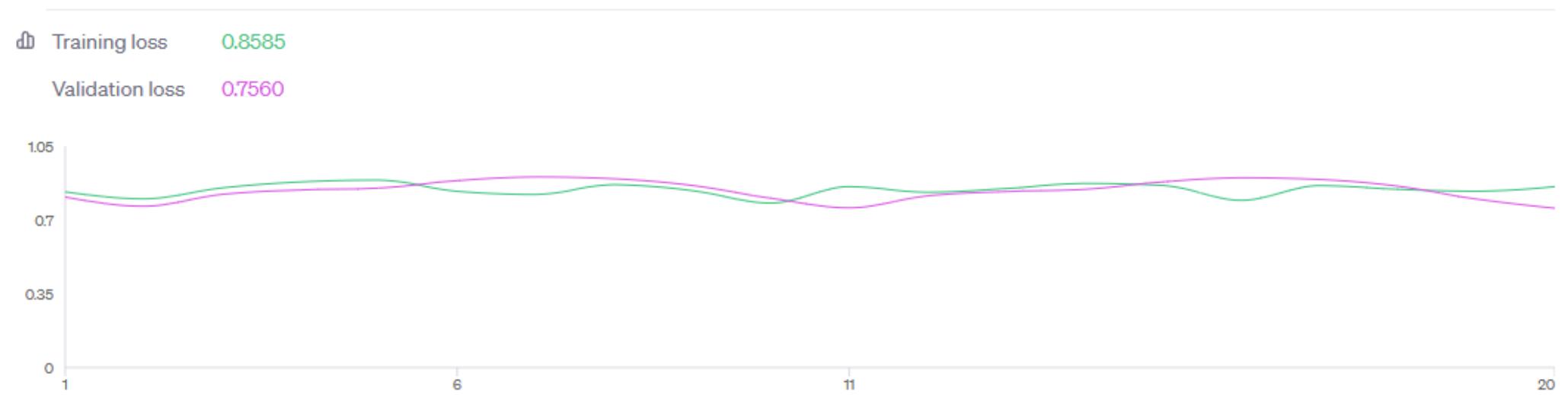
- system content นั้นจะกำหนด style การตอบด้วยคำว่า "คุณเป็นโภค์นำทางที่เป็นมิตรและสุภาพ" ในทุกๆ messages เพื่อกำหนด tone และ style แบบเดียวกัน
- user content นั้นจะมีการกำหนดรูปแบบคำถ้าที่เหมือนกันโดยจะเป็นในรูปแบบ "ฉันต้องการข้อมูลรายละเอียดเกี่ยวกับ<ชื่อสถานที่>"
- assistant content หรือคำตอบที่ต้องการให้โมเดลตอบจะใส่รายละเอียดข้อมูลของสถานที่นั้นไปเป็น long text ยาว ๆ และนำ dataset ดังกล่าวไปเทรนกับ gpt-3.5-turbo-1106

1

ทดสอบ Fine-tuning Dataset ที่แตกต่างกัน

Dataset แบบที่ 1

```
{"messages": [  
    {"role": "system", "content": "คุณเป็นไกด์นำทางที่เป็นมิตรและสุภาพ"},  
    {"role": "user", "content": "ฉันต้องการข้อมูลรายละเอียดเกี่ยวกับม่อนแจ่ม"},  
    {"role": "assistant", "content": "ม่อนแจ่ม เป็นประเภท สถานที่ท่องเที่ยวโดยและภูเขา - ข้อมูล"}  
]
```



1

ทดสอบ Fine-tuning Dataset ที่แตกต่างกัน

โมเดลที่ใช้ : gpt-3.5-turbo-1106

```
model1 = openai.fine_tuning.jobs.create(  
    training_file="file-lUVKIZYrqECiJs23499L5a7T",  
    validation_file="file-LM6xSQnLCerJ846jS3K1Y6VM",  
    model="gpt-3.5-turbo-1106",  
    suffix="gpt3.5turbomodel1",  
    hyperparameters={  
        "n_epochs": 3,  
        "batch_size": 8,  
        "learning_rate_multiplier": 0.01  
    }  
)
```

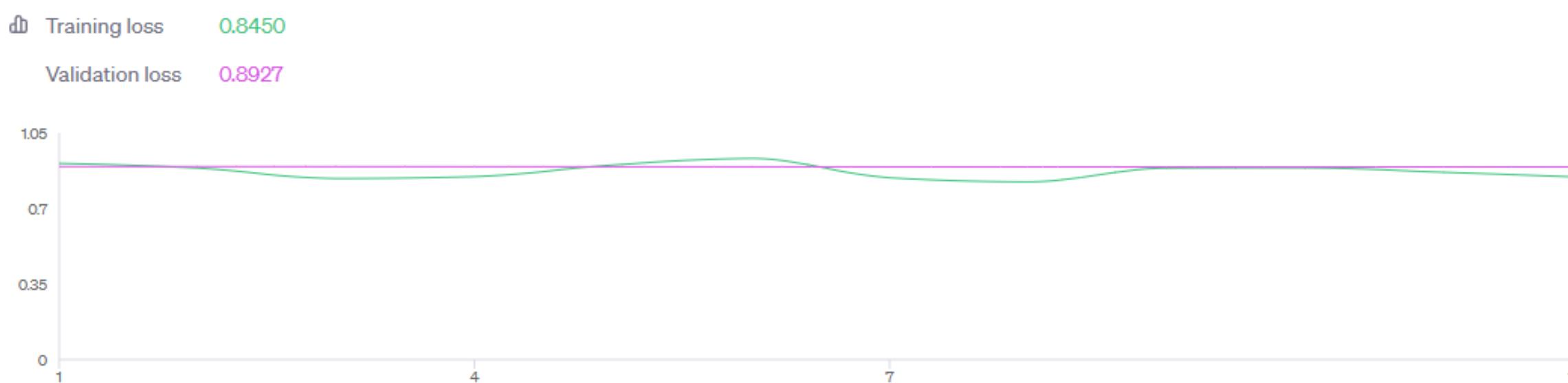
Dataset แบบที่สองมีจำนวน data อยู่ที่ 40 samples

- system content นี้จะเปลี่ยนเป็นคำว่า "You are a friendly and polite guide who give the travel place information in Thailand."
- user content ปรับเปลี่ยนให้มี pattern มากรขึ้นตามตัวอย่างที่ได้พบ
- assistant content หรือคำตอบที่ต้องการให้โมเดลตอบจะใส่รายละเอียดข้อมูลของสถานที่นั้นไปเป็น long text เช่นเดิม

1

ทดสอบ Fine-tuning Dataset ที่แตกต่างกัน

Dataset แบบที่ 2



1

ทดสอบ Fine-tuning Dataset ที่แตกต่างกัน

คำตอบของ dataset กึ่ง 2 แบบ

```
1 response = openai.chat.completions.create(  
2   model="ft:gpt-3.5-turbo-1106:personal:gpt3-5turbomodel1:8MY0VlsQ",  
3   temperature=0,  
4   max_tokens=1000,  
5   messages=[  
6     {"role": "system", "content": "คุณเป็นไกด์นาทางที่มีนิตรและสุภาพ"},  
7     {"role": "user", "content": "ฉันต้องการข้อมูลรายละเอียดเกี่ยวกับม่อนแจ่ม"}  
8   ]  
9 )  
10 print(response.choices[0].message)
```

```
1 response = openai.chat.completions.create(  
2   model="ft:gpt-3.5-turbo-1106:personal:gpt3-5turbomodel2:8McL0LVJ",  
3   temperature=0,  
4   max_tokens=1000,  
5   messages=[  
6     {"role": "system", "content": "You are a friendly and polite guide who give the travel place information in Thailand."},  
7     {"role": "user", "content": "ชื่อสถานที่: ม่อนแจ่ม\ก\ประเทศไทย สถานที่ท่องเที่ยวโดยและภูเขา\ก\ข้อมูลรายละเอียดสถานที่: "}]  
8   ]  
9 )  
10 print(response.choices[0].message)
```

คำตอบที่ได้: “ม่อนแจ่ม (Mon Cham) เป็นสถานที่ท่องเที่ยวที่ตั้งอยู่ในอำเภอแม่ริม จังหวัดเชียงใหม่ ประเทศไทย ม่อนแจ่มเป็นที่รู้จักด้วยกิวักศน์ที่สวยงามของภูเขาและทุ่งนาที่ล้อมรอบ นอกจากนี้ยังมีสวนผลไม้และสวนดอกไม้ที่สวยงามอีกด้วย หากที่ม่อนแจ่มนั้นบักท่องเที่ยวสามารถเดินเล่นรอบเขารือทำกิจกรรมทำการเกษตร เช่น การเก็บผลไม้ หรือดูดดอกไม้ได้ นอกจากนี้ยังมีร้านอาหารและรีสอร์ฟที่สามารถพักผ่อนและพักค้างคืนได้ หากการเดินทางไปยังม่อนแจ่มสามารถใช้รถยนต์ส่วนตัวหรือจักรยานจากเชียงใหม่ โดยมีระยะทางประมาณ 40 กิโลเมตร และเส้นทางที่สวยงามและสะดวกสบาย”

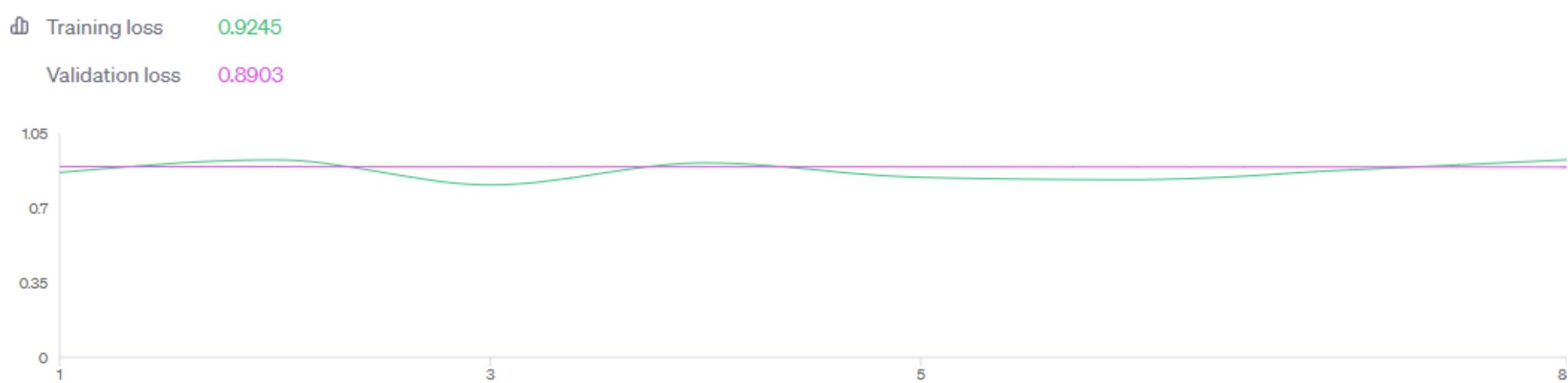
คำตอบที่ได้: “สวัสดีครับ/ค่ะ! สถานที่ท่องเที่ยว “ม่อนแจ่ม” เป็นสถานที่ท่องเที่ยวที่ตั้งอยู่บนเทือกเขาดอยอ่างขางในจังหวัดเชียงใหม่ ม่อนแจ่มเป็นจุดชมวิวที่สวยงามที่สุดของดอยอ่างขาง ที่นี่มีกิจกรรมที่สวยงามของทะเลหมอกและภูเขาที่สลัดใส นอกจากนี้ยังมีสวนดอกไม้ที่สวยงามและเป็นที่รู้จักอีกด้วย นอกจากนี้ยังมีกิจกรรมต่างๆ เช่น การปีนเขา การเดินทางด้วยจักรยาน และการเดินเล่นในสวนดอกไม้ ที่นี่ยังมีร้านอาหารและร้านขายของที่จำพวกผลผลิตภูเขาให้บริการด้วยครับ/ค่ะ ถ้าคุณชอบการผจญภัยและการสัมผัสรธรรมชาติ ม่อนแจ่มเป็นสถานที่ท่องเที่ยวที่น่าสนใจมากครับ/ค่ะ หวังว่าข้อมูลนี้จะมีประโยชน์สำหรับคุณครับ/ค่ะ ถ้ามีข้อสงสัยเพิ่มเติม กรุณาสอบถามเพิ่มเติมครับ/ค่ะ”

2

nodejs ทดสอบ Fine-tuning โมเดลที่ Hyperparameters แต่กต่างกัน

แบบที่ 1

```
model3 = openai.fine_tuning.jobs.create(  
    training_file="file-co6ZMnWyGygwbWNFCedz3hi8",  
    validation_file="file-EsII02TMtG7ZQhTB3AtmCpct",  
    model="gpt-3.5-turbo-1106",  
    suffix="gpt3.5turbomodel3",  
    hyperparameters={  
        "n_epochs": 5,  
        "batch_size": 8,  
        "learning_rate_multiplier": 0.01  
    }  
)
```

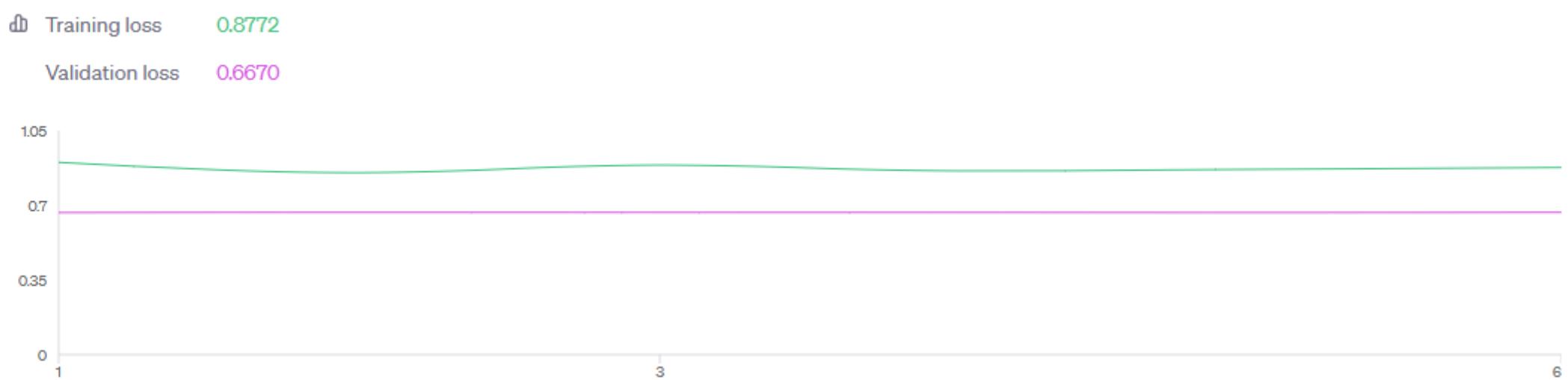


2

ทดสอบ Fine-tuning โมเดลที่ Hyperparameters แต่กต่างกัน

แบบที่ 2

```
model4 = openai.fine_tuning.jobs.create(  
    training_file="file-co6ZMnWyGygwbWNFCedz3hi8",  
    validation_file="file-EsII02TMtG7ZQhTB3AtmCpct",  
    model="gpt-3.5-turbo-1106",  
    suffix="gpt3.5turbomodel3",  
    hyperparameters={  
        "n_epochs": 3,  
        "batch_size": 16,  
        "learning_rate_multiplier": 0.001  
    }  
)
```

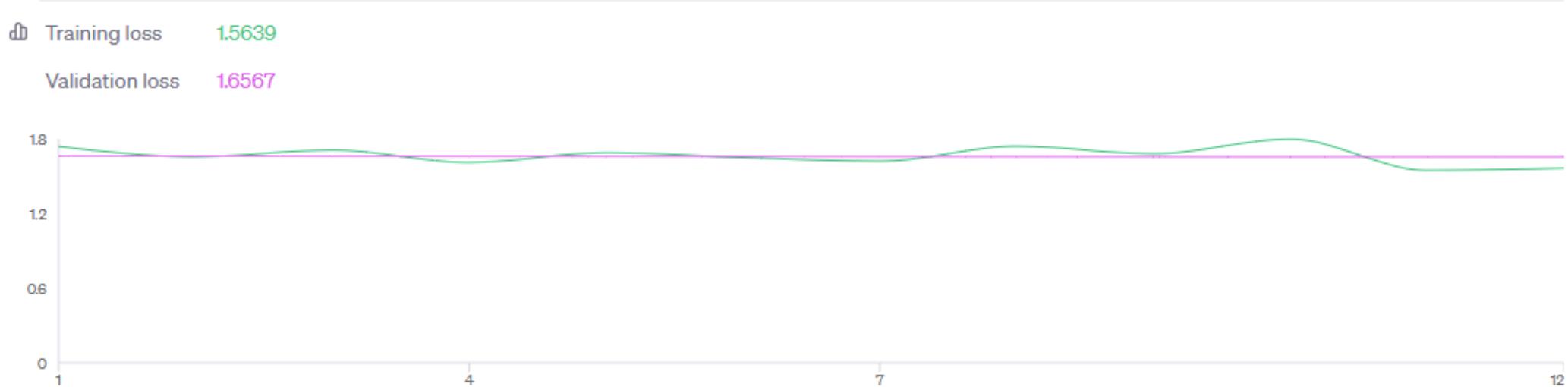


3

nodejs ทดสอบ Fine-tuning Pre-trained Model ที่แตกต่างกัน

โมเดลที่ใช้ Pre-trained model: babbage-002

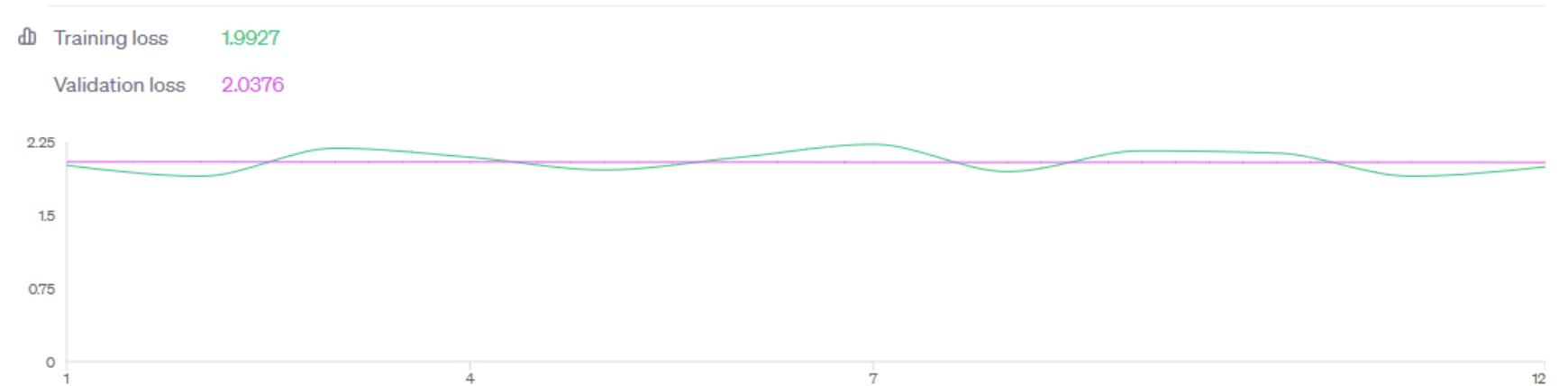
```
model5 = openai.fine_tuning.jobs.create(  
    training_file="file-9NXUyvYYObtIdZDQWDsi6INX",  
    validation_file="file-hq51dEn60webj6sjTXCfHZTz",  
    model="babbage-002",  
    suffix="babagemode1",  
    hyperparameters={  
        "n_epochs": 3,  
        "batch_size": 8,  
        "learning_rate_multiplier": 0.01  
    }  
)
```



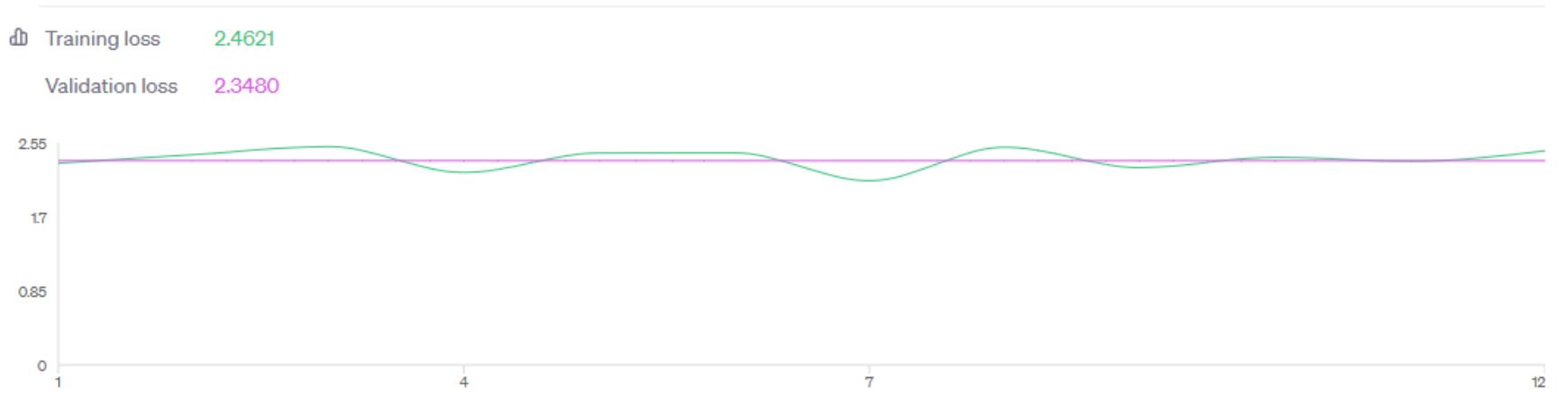
4

node 4: Fine-tuning Dataset គ្មានភាសា

ໂນໂດລក់ໃច្ច Pre-trained model: gpt-3.5-turbo



ໂນໂດລក់ໃច្ច Pre-trained model: babbage-002



Conclusion

จากการทดสอบทั้ง 4 อย่างพบว่าผลลัพธ์ไม่ได้เป็นไปอย่างที่คาดหวังไว้เนื่องจากคุณภาพของ dataset ยังดีไม่นักพوรวมไปถึงขนาดของข้อมูลที่มีขนาดน้อยเกินไป ดูได้จากผลการทดสอบที่ 1 ที่ลองเปลี่ยนรูปแบบของ dataset พบร่วมกับ output ที่ได้นั้นดีขึ้นในเชิงของการเรียบเรียงประโยคแต่ก็ยังไม่ใช่คำตอบที่มาจากการทดสอบที่ 2 และ 3 ที่ลองเปลี่ยน hyperparameters หรือ pre-trained model ก็ยังไม่สามารถเรียนรู้ dataset ได้เช่นเดียวกันจนมาถึงการทดสอบที่ 4 ที่ลองเปลี่ยนเป็นภาษาอังกฤษซึ่งทำให้มี token น้อยลงแต่ค่า loss กับเพิ่มขึ้นและไม่ได้เรียนรู้ dataset

