Generative AI, OpenAI API กับ Python
สร้างแอปล้ำ ๆ จาก 0 ถึงโปร

โดย

Nextcodecamp

จัดทำโดย
รองศาสตราจารย์ ดร. กฤดาภัทร สีหารี
19/76 ซอยแจ้งวัฒนะ-ปากเกร็ด 41
ตำบลคลองเกลือ อำเภอปากเกร็ด
จังหวัดนนทบุรี 11120

สงวนสิทธิ์ตามพระราชบัญญัติลิขสิทธิ์ 2537
ห้ามคัดลอกเนื้อหาก่อนได้รับอนุญาต

ข้อมูลทางบรรณานุกรมหนังสือ
กฤดาภัทร สีหารี,
Generative AI, OpenAI API กับ Python สร้างแอปล้ำ ๆ จาก 0 ถึงโป
2568.
189 หน้า.
ISBN (e-book): 978-616-623-347-6
1. Generative AI 2. ChatGPT 3. OpenAI API

หนังสือ E-Book จัดจำหน่ายออนไลน์

คำนำ

กระแส Generative AI มาถึงตั้งนานแล้วแต่ผู้เขียนเพิ่งได้สัมผัสจริงจัง ลองใช้ ChatGPT หาข้อมูลช่วยเขียนโค้ด จึงเกิดคำถามว่ามาถึงยุคที่ไม่จำเป็นต้องเรียนรู้การเขียนโปรแกรมแล้วหรือ แต่สิ่งหนึ่งที่ปฏิเสธไม่ได้คือ มันทำให้ชีวิตง่ายขึ้น มีเพื่อนช่วยคิด มีเพื่อนช่วยเขียน และมีเพื่อนช่วยจัดการ ในความคิดเห็นส่วนตัวคงไม่เห็นด้วยว่าไม่จำเป็นต้องเขียนโปรแกรมเป็น เพราะโค้ดที่ได้มาจาก ChatGPT นั้นอาจจะยังไม่ดีพอ อีกอย่างระบบ Generative AI ต่างก็มี API ให้เขียนโปรแกรมติดต่อเพื่อทำงานกับเขาได้ นี่แสดงให้เห็นว่าการเขียนโปรแกรมยังมีความจำเป็นเพียงแค่ไม่จำเป็นต้องทำเองหมดทุกอย่างแต่ต้องรู้จักนำส่วนต่าง ๆ มาทำงานร่วมกันเพื่อสร้างสรรค์สิ่งใหม่ ๆ ที่นำไปใช้ประโยชน์ได้ การปรับตัวเพื่อทำงานร่วมกันกับ AI จึงเป็นสิ่งที่ท้าทายไม่ว่าสาขาอาชีพใดก็จำเป็นต้องปรับตัวเพื่อใช้มันให้เป็น

หนังสือเล่มนี้แนะนำการใช้ประโยชน์จาก ChatGPT เรียนรู้การทำงานพื้นฐานใน ChatGPT Platform และการทำงานพื้นฐานใน OpenAI Platform ผู้เขียนพยายามถ่ายทอดให้ผู้ที่สนใจได้เรียนรู้ และนำไปใช้งานได้จริงไม่ว่าผู้อ่านจะมีหรือไม่มีความรู้เชิงเทคนิคก็สามารถอ่านได้ ผู้อ่านจะเห็นศักยภาพของ Generative AI ในด้านต่าง ๆ และมองเห็นประโยชน์ในการนำไปใช้ในชีวิตประจำวัน อีกทั้งได้ยกตัวอย่างการเขียน AI แอปพลิเคชันแบบ No code หรือไม่เขียนโค้ด และแบบ Low code คือเขียนโค้ดเล็กน้อย อีกทั้งยกตัวอย่างการใช้งาน ChatGPT เพื่อประโยชน์ในด้านต่าง ๆ ผู้เขียนหวังเป็นอย่างยิ่งว่าหนังสือเล่มนี้จะทำให้ผู้อ่านเข้าใจ และใช้ประโยชน์จาก Generative AI ได้อย่างมีประสิทธิภาพต่อไป

รศ.ดร.กฤดาภัทร สีหารี
29 มีนาคม 2568

สารบัญ

คำนำ ii
สารบัญ iii
สารบัญภาพ iv
บทที่ 1 แนะนำหนังสือและ Generative AI **1**
1.1 แนะนำหนังสือ 1
1.2 AI กับ Generative AI 2
1.2.1 การเรียนรู้ของเครื่องจักร 2
1.2.2 การเรียนรู้เชิงลึก 4
1.2.3 Generative AI 6
1.3 LLMs 8
1.3.1 LLMs กับการประมวลผลข้อความ 8
1.3.2 สถาปัตยกรรมของ LLMs 10
1.3.3 เอมเบดดิง 11
1.3.4 การสร้างประโยคแบบ Nucleaus sampling 14
บทที่ 2 ความสามารถทั่วไปของ ChatGPT **16**
2.1 OpenAI กับ ChatGPT 16
2.2 ความสามารถทั่วไปของ ChatGPT 17
2.3 ChatGPT กับการค้นหาข้อมูล 20
2.4 ChatGPT กับการสื่อสาร 25
2.5 ChatGPT กับการเขียนโปรแกรมและกราฟ 26
2.6 ChatGPT กับการวิเคราะห์ข้อมูล 28

2.7 ChatGPT กับการประมวลผลภาษาธรรมชาติ 31
2.8 ChatGPT กับการสร้างภาพ 34
2.9 ChatGPT กับการแปลงเสียงเป็นข้อความ 38
2.10 ChatGPT กับการจัดการงานในออฟฟิต 40
2.10.1 เขียนจดหมาย 40
2.10.2 สร้างสไลด์ PowerPoint 42
2.10.3 จัดการตาราง Excel 44
2.10.4 หาสูตรคำนวณ 46
2.10.5 พรอมต์เขียนโค้ด VBA สร้างมาโคร 46
2.11 เปรียบเทียบ ChatGPT กับ ChatGPT Plus และ OpenAI API 50
บทที่ 3 เทคนิคเขียนพรอมต์ 53
3.1 พรอมต์ 53
3.2 องค์ประกอบพรอมต์ที่ดี 56
3.2.1 ตัวอย่างของพรอมต์ที่มีเฉพาะคำสั่ง 56
3.2.2 ตัวอย่างของพรอมต์ที่มีคำสั่งและบริบท 57
3.2.3 ตัวอย่างของพรอมต์ที่มีคำสั่งและบริบทที่เจาะจงขึ้น 57
3.2.4 ตัวอย่างของพรอมต์ที่มีคำสั่ง, บริบท และเอาต์พุต 59
3.2.5 ตัวอย่างของพรอมต์ที่มีคำสั่ง, บริบท เอาต์พุต และรูปแบบ 59
3.3 พรอมต์แบบ Zero-shot, One-shot และ Few-shot 61
3.4 พรอมต์ที่มีบุคลิกภาพเฉพาะตัว 62
3.5 พรอมต์ทำงานทีละขั้นตอน 64
3.6 พรอมต์กับ Markdown 66
3.7 แหล่งสะสมพรอมต์ 68
3.8 การปรับแต่ง ChatGPT 70

บทที่ 4 รู้จัก API กับ OpenAI API **73**
4.1 API คืออะไร 73
4.2 องค์ประกอบของ HTTP/HTTPS Request 74
4.3 JSON 75
4.3.1 รูปแบบในการแสดงข้อมูล 75
4.3.2 ตัวอย่างการอ้างอิงข้อมูลใน JSON 77
4.4 ChatGPT Platform กับ OpenAI Platform 78
4.5 ประโยชน์และการคิดค่าบริการใน OpenAI 79
4.6 การตั้งค่าเงื่อนไขในการตอบกลับ 86
4.7 ทดสอบ OpenAI API กับ Postman 90
4.7.1 สร้างโปรเจ็กต์ 90
4.7.2 สร้าง API Key 91
4.7.3 ทดสอบการเรียกใช้ OpenAI API ใน Postman 93
4.8 ติดตั้ง OpenAI 98
บทที่ 5 พัฒนา AI แอปพลิเคชันกับ OpenAI Platform **99**
5.1 AI แอปพลิเคชัน 99
5.2 การพัฒนาแอปพลิเคชันด้วย Chat Completions API 99
5.2.1 คำร้องขอ Chat Completions API 101
5.2.2 ตัวอย่างการสร้าง Chatbot อย่างง่ายที่จดจำอินพุต 103
5.2.3 ตัวอย่างการสรุปข้อความ 106
5.2.4 ตัวอย่างแอปพลิเคชันเขียนโค้ด 108
5.3 ผู้ช่วย 110
5.3.1 การสร้างผู้ช่วยใน Playground 112
5.3.2 การสร้างผู้ช่วยด้วย OpenAI API 115

5.3.3 ตัวอย่าง AI Multi-agent systems 119
5.4 การเรียกฟังก์ชัน 124
5.4.1 การสร้างฟังก์ชันที่ Playground 125
5.4.2 ข้อกำหนดฟังก์ชัน 128
5.4.3 การสร้างฟังก์ชันด้วย OpenAI API 132
5.5 การนับโทเค็นและชังค์ 137
บทที่ 6 การสร้าง Custom GPT 141
6.1 Custom GPT คืออะไร 141
6.2 ระบบนิเวศ OpenAI และ Custom GPT 142
6.3 สร้าง Custom GPT ใน ChatGPT (Plus) 146
6.3.1 สร้าง Custom GPT 146
6.3.2 การประกาศ Custom GPT 149
6.3.3 การสร้าง Action ใน Custom GPT 151
6.3.4 การสร้าง Action ที่มีการยืนยันตัวตนใน Custom GPT 158
6.4 Open API schema 161
บทที่ 7 การพัฒนา AI แอปพลิเคชัน 164
7.1 การพัฒนา AI แอปพลิเคชัน 164
7.2 แอปพลิเคชัน Insurance Buddy 165
7.3 แอปพลิเคชันแนะนำตำแหน่งงาน 168
7.4 แอปพลิเคชันสอบถามระเบียบการรับสมัครเข้าศึกษา 169
7.5 แอปพลิเคชันสรุปข้อความจากเอกสาร PDF 171
7.6 แอปพลิเคชันผู้ช่วยสรุปข้อความจากเสียงพูด 174
7.7 สร้างแอปพลิเคชันผู้ช่วยหาข้อผิดพลาดของโค้ด 177
7.8 แอปพลิเคชันระบบถามตอบที่มีการปรับปรุงคำตอบจากโดเมน 181

7.8.1 การพัฒนาด้วยการสร้าง Custom GPT 181
7.8.2 การพัฒนาด้วยการสร้างผู้ช่วยที่ OpenAI Platform 181
7.8.3 การพัฒนาด้วย OpenAI API 182
7.9 พัฒนาแอปพลิเคชันเพื่อเรียกใช้ OpenAI API 187
7.9.1 สร้าง Custom GPT เรียกใช้ Chat Completions API 187
7.9.2 พัฒนาแอปพลิเคชันสร้างภาพ 189
7.10 พัฒนาเว็บแอปพลิเคชัน Chat คุยกับ ChatGPT 193
7.11 การประสาน ChatGPT ในแอปพลิเคชันต่าง ๆ 195
7.12 สรุป 197
บทที่ 8 เรื่องอื่น ๆ **198**
8.1 ChatGPT Task 198
8.2 LangChain 200
8.2.1 การทำงานกับ LangChain 200
8.2.2 ตัวอย่างการสร้าง AI แอปพลิเคชันด้วย LangChain 201
8.3 ตัวอย่างการประยุกต์ใช้ AI ในระบบงานต่าง ๆ 205
8.3.1 Copilot 205
8.3.2 Postman 206
8.3.3 Adobe Firefly 208
8.3.4 Hostinger 209
8.4 สรุป 210
บรรณานุกรม **212**

สารบัญภาพ

ภาพที่ 1.1 การเรียนรู้ของเครื่องจักร 3
ภาพที่ 1.2 เทคนิคการเรียนรู้ด้วยเครื่องจักร 4
ภาพที่ 1.3 การเรียนรู้เชิงลึก 5
ภาพที่ 1.4 Generative AI 7
ภาพที่ 1.5 การจำแนกโมเดล Generative AI [1] 7
ภาพที่ 1.6 Text, Tokenization, Token, Embedding, Vector 9
ภาพที่ 1.7 สถาปัตยกรรม LLM [4] 10
ภาพที่ 1.8 Cosine similarity 12
ภาพที่ 2.1 กล่องป้อนคำสั่งหรือพรอมต์ 17
ภาพที่ 2.2 ฟีเจอร์ต่าง ๆ 18
ภาพที่ 2.3 ฟีเจอร์ 18
ภาพที่ 2.4 แก้ไขโค้ดใน Canvas 19
ภาพที่ 2.5 Inpainting 20
ภาพที่ 2.6 ค้นหาข้อมูลในเว็บ 21
ภาพที่ 2.7 ผลลัพธ์จากการค้นหา 21
ภาพที่ 2.8 ผลลัพธ์จากการค้นหา 22
ภาพที่ 2.9 ผลลัพธ์จากการค้นหาด้วยการเลือก Deep research 23
ภาพที่ 2.10 รายการเอกสารที่สืบค้นจากฐานข้อมูลงานวิจัย 23
ภาพที่ 2.11 การเน้นข้อความที่ ChatGPT วิเคราะห์ 24
ภาพที่ 2.12 พรอมต์กำหนดบทบาทของระบบ 25
ภาพที่ 2.13 ภาพกระเป๋าที่ออกแบบ 26

ภาพที่ 2.14 การสร้างโค้ดและวาดกราฟ 27
ภาพที่ 2.15 กราฟปรับแกน x-axis 27
ภาพที่ 2.16 ข้อมูลรีวิวร้านอาหาร 28
ภาพที่ 2.17 พรอมต์ถามการวิเคราะห์ข้อมูล 29
ภาพที่ 2.18 ผลการวิเคราะห์ข้อมูล 29
ภาพที่ 2.19 พรอมต์คำสั่งวิเคราะห์ข้อมูล 30
ภาพที่ 2.20 ผลการวิเคราะห์ข้อมูล 31
ภาพที่ 2.21 เติมคำลงในช่องว่าง 32
ภาพที่ 2.22 เติมคำลงในช่องว่าง 32
ภาพที่ 2.23 การอนุมาน 33
ภาพที่ 2.24 โฆษณาขายเสื้อยืด 33
ภาพที่ 2.25 บล็อกรีสอร์ท 34
ภาพที่ 2.26 บ้านสไตล์ Tiny house 35
ภาพที่ 2.27 ภาพบ้านที่ปรับปรุงโดยซ่อนบันได 35
ภาพที่ 2.28 วิเคราะห์บ้าน 36
ภาพที่ 2.29 พรอมต์วิเคราะห์ภาพ 36
ภาพที่ 2.30 พรอมต์วิเคราะห์ภาพ 37
ภาพที่ 2.30 TTS ใน Playground 38
ภาพที่ 2.31 TTS 39
ภาพที่ 2.32 ร่างจดหมายลาป่วย 40
ภาพที่ 2.33 ร่างจดหมายลาออก 41
ภาพที่ 2.34 สร้างหัวข้อการนำเสนอ 43
ภาพที่ 2.35 พรอมต์สร้างโค้ด VBA 43
ภาพที่ 2.36 วางโค้ด VBA เพื่อสร้างสไลด์ Powerpoint 44

ภาพที่ 2.37 พรอมต์สร้างไฟล์ Excel 45
ภาพที่ 2.38 พรอมต์สร้างไฟล์ Excel 46
ภาพที่ 2.39 พรอมต์ขอให้เขียนโค้ด 47
ภาพที่ 2.40 โค้ด VBA 47
ภาพที่ 2.41 พรอมต์สั่งให้ปรับปรุงโค้ด 48
ภาพที่ 2.42 สร้างมาโครจากโค้ด ChatGPT 49
ภาพที่ 2.43 ใบเรียกเก็บเงิน 49
ภาพที่ 3.1 อินพุตรับพรอมต์ใน ChatGPT 53
ภาพที่ 3.2 โทเค็นกับแคช 54
ภาพที่ 3.3 อินพุตรับพรอมต์ใน Playground (OpenAI Platform) 55
ภาพที่ 3.4 อินพุตรับพรอมต์ใน Playground (OpenAI Platform) 55
ภาพที่ 3.5 พรอมต์ถามชื่อแม่น้ำ 56
ภาพที่ 3.6 พรอมต์ที่มีคำสั่งและบริบท 57
ภาพที่ 3.7 พรอมต์มีคำสั่งและบริบทเจาะจง 58
ภาพที่ 3.8 พรอมต์มีคำสั่ง บริบทเจาะจง และเอาต์พุต 59
ภาพที่ 3.9 พรอมต์มีคำสั่ง บริบทเจาะจง เอาต์พุต และรูปแบบ 60
ภาพที่ 3.10 พรอมต์ One-shot 61
ภาพที่ 3.11 พรอมต์ One-shot (ต่อ) 62
ภาพที่ 3.12 พรอมต์ที่มีบุคลิกภาพ 63
ภาพที่ 3.13 พรอมต์ที่มีบุคลิกภาพ (ต่อ) 63
ภาพที่ 3.14 พรอมต์ที่มีบุคลิกภาพ 64
ภาพที่ 3.15 พรอมต์สั่งงานทีละขั้นตอน 65
ภาพที่ 3.16 พรอมต์แก้ไขระหว่างทำงานทีละขั้นตอน 66
ภาพที่ 3.17 พรอมต์สั่งเขียน Markdown 66

ภาพที่ 3.18 บล็อกในรูปแบบ Markdown 67
ภาพที่ 3.19 บล็อกใน VS Code 68
ภาพที่ 3.20 ตัวอย่างเว็บไซต์ Flowgpt 69
ภาพที่ 3.21 เพจของผู้นำเสนอปกหนังสือ 69
ภาพที่ 3.22 ปรับแต่งสไตล์การตอบกลับของพรอมต์ 70
ภาพที่ 3.23 สั่งพรอมต์ให้สร้างพรอมต์ 71
ภาพที่ 3.24 แก้ไขสไตล์ของพรอมต์ 71
ภาพที่ 3.25 การตอบคำถามตามสไตล์ของพรอมต์ 72
ภาพที่ 4.1 API กับ OpenAI Platform 73
ภาพที่ 4.2 API Endpoint และ HTTP Address 75
ภาพที่ 4.3 อัตราค่าบริการการเรียกใช้ API 80
ภาพที่ 4.4 OpenAI Platform 81
ภาพที่ 4.5 หน้าจอหลักเมื่อเข้าสู่ระบบแล้ว 82
ภาพที่ 4.6 หน้าจอ Playground 82
ภาพที่ 4.7 หน้าจอ Dashboard 83
ภาพที่ 4.8 หน้าจอ API reference 83
ภาพที่ 4.9 การตอบกลับ 85
ภาพที่ 4.10 การตอบกลับ 86
ภาพที่ 4.11 หน้าจอ API reference 90
ภาพที่ 4.12 สร้างโปรเจ็กต์ใหม่ 91
ภาพที่ 4.13 API keys 91
ภาพที่ 4.14 กำหนดชื่อ API key 92
ภาพที่ 4.15 API key 92
ภาพที่ 4.16 OpenAI API ผู้จัดทำคือทีมงาน Postman 93

ภาพที่ 4.17 Fork OpenAI API 94
ภาพที่ 4.18 กำหนด Workspace เพื่อวาง OpenAI API ใช้งานส่วนตัว 94
ภาพที่ 4.19 ส่งคำร้องขอไปที่ Chat Completions API 95
ภาพที่ 4.20 กำหนด OpenAI API Key 96
ภาพที่ 4.21 ทดสอบคำร้องขอไปที่ Chat Completions API 96
ภาพที่ 4.22 รันสคริปต์ทดสอบการทำงานของ OpenAI API 97
ภาพที่ 5.1 การสื่อสารเมื่อทำงานกับผู้ช่วย 111
ภาพที่ 5.2 ผู้ช่วยตอบคำถามในการซ่อมรถยนต์ 112
ภาพที่ 5.3 ผู้ช่วยตอบคำถามในการซ่อมรถยนต์ 113
ภาพที่ 5.4 ทักทายผู้ช่วยตอบคำถามในการซ่อมรถยนต์ 113
ภาพที่ 5.5 ถามผู้ช่วยเกี่ยวกับบริษัทประกันภัยรถยนต์ 114
ภาพที่ 5.6 ระบบเอเจนต์ 120
ภาพที่ 5.7 การส่งข้อความ 121
ภาพที่ 5.8 การตัดสินใจในการสอบถาม 125
ภาพที่ 5.9 การตัดสินใจของระบบถามไปยังฟังก์ชันผู้ใช้ 126
ภาพที่ 5.10 ตัวอย่าง Function schema ของ get_weather 127
ภาพที่ 5.11 ตัวอย่าง Function schema ของ get_weather ปรับใหม่ 127
ภาพที่ 5.12 การตัดสินใจของระบบถามไปยังโมเดล 128
ภาพที่ 5.13 พรอมต์สร้าง Function schema 129
ภาพที่ 5.14 Function schema ฟังก์ชัน get_contact_details 130
ภาพที่ 5.15 ฟังก์ชัน get_contact_details ถูกเรียก 131
ภาพที่ 5.16 ChatGPT ปฏิเสธการเข้าถึงข้อมูลของหน่วยงานรัฐ 131
ภาพที่ 5.17 เครื่องมือคำนวณโทเค็น 137
ภาพที่ 6.1 Custom GPT 142

ภาพที่ 6.2 Custom GPT 143
ภาพที่ 6.3 Custom GPT ของ ChatGPT 143
ภาพที่ 6.4 โปรไฟล์ Custom GPT ของ Expedia 144
ภาพที่ 6.5 การติดต่อ Third party 144
ภาพที่ 6.6 หน้าต่างแรกของ Expedia (Custom GPT) 145
ภาพที่ 6.7 แช็ตกับ Expedia 145
ภาพที่ 6.8 หน้าจอสร้าง Custom GPT 146
ภาพที่ 6.9 สร้าง Custom GPT ชื่อ Computer Science QA 147
ภาพที่ 6.10 ออกแบบโปรไฟล์ Custome GPT 147
ภาพที่ 6.11 ตั้งค่าหน้าต่างแรกของ Custom GPT 148
ภาพที่ 6.12 เลือกความสามารถของ GPT ที่จะใช้ 148
ภาพที่ 6.13 ค้นหา Custom GPT ที่ประกาศ/ลงทะเบียนแล้ว 149
ภาพที่ 6.14 Custom GPT แก้ไขคำถามนำทางแช็ต 149
ภาพที่ 6.15 หมวดหมู่ของ Custom GPT 150
ภาพที่ 6.16 สำเนาลิงค์ไปแชร์ 150
ภาพที่ 6.17 Favorite Grocery Products 151
ภาพที่ 6.18 ทดสอบ Action เรียกฟังก์ชัน 152
ภาพที่ 6.19 การตอบกลับหลังเรียกฟังก์ชันแล้ว 153
ภาพที่ 6.20 สร้าง Schema ของ Custom GPT 154
ภาพที่ 6.21 Action GPT สร้าง Function schema/Open API schema 154
ภาพที่ 6.22 ตรวจสอบการทำงานของ API ด้วย Postman 155
ภาพที่ 6.23 Open API schema ในรูปแบบ yaml 156
ภาพที่ 6.24 กำหนด Schema ให้กับ Action 157
ภาพที่ 6.25 ทดสอบการเรียกฟังก์ชัน 158

ภาพที่ 6.26 ตั้งค่า Custom GPT 159
ภาพที่ 6.27 กำหนดวิธียืนยันตัวตนในการติดต่อกับ API 159
ภาพที่ 6.28 API Key ใน API ที่สอดคล้องกับวิธี Custom ใน OpenAI API 160
ภาพที่ 6.29 Custom GPT ติดต่อกับ API 160
ภาพที่ 7.1 รายการเอกสารตั้งเบิก 165
ภาพที่ 7.2 เอกสารอนุมัติการเบิก 166
ภาพที่ 7.3 เอกสารถูกนำไปสร้างเวกเตอร์เอมเบดดิง 166
ภาพที่ 7.4 สอบถามเอกสาร 167
ภาพที่ 7.5 สอบถามการอนุมัติการเบิกค่าสินไหม 167
ภาพที่ 7.6 ตำแหน่งงานและทักษะ 168
ภาพที่ 7.7 แช็ตสอบถามตำแหน่งที่เหมาะสมกับทักษะผู้สมัคร 168
ภาพที่ 7.8 สอบถามการรับสมัคร 169
ภาพที่ 7.9 สอบถามการรับสมัครแต่ไม่สามารถหาคำตอบได้ 170
ภาพที่ 7.10 การทำงานการสรุปข้อความ 171
ภาพที่ 7.11 การทำงานการถอดข้อความจากเสียง 174
ภาพที่ 7.12 การทำงานในผู้ช่วย 181
ภาพที่ 7.13 ผู้ช่วยสอบถามระบบถามตอบ 182
ภาพที่ 7.14 การทำงานในแอปพลิเคชันถามตอบ 182
ภาพที่ 7.15 qa_chatgpt.csv 183
ภาพที่ 7.16 Custom GPT เรียก Chat Completions API 188
ภาพที่ 7.17 กำหนด OpenAI API Key 188
ภาพที่ 7.18 การเรียก API ใน Custom GPT 189
ภาพที่ 7.19 ส่วนขยาย Apps Script 195
ภาพที่ 7.20 อิดิเตอร์ Apps Script 196

ภาพที่ 7.21 แช็ตกับ ChatGPT ใน Google Sheets 196
ภาพที่ 7.22 การประสานงานระหว่างแอปพลิเคชันกับ OpenAI API 197
ภาพที่ 8.1 ChatGPT Tasks 198
ภาพที่ 8.2 ChatGPT Tasks เมนู 199
ภาพที่ 8.3 แก้ไขการแจ้งเตือนส่งอีเมล 199
ภาพที่ 8.4 LangChain 200
ภาพที่ 8.5 แช็ต 202
ภาพที่ 8.6 Copilot 205
ภาพที่ 8.7 Sign in 206
ภาพที่ 8.8 Ask Copilot 206
ภาพที่ 8.9 Postbot ใน Postman 207
ภาพที่ 8.10 Test case 208
ภาพที่ 8.11 สร้างภาพจากข้อความใน Firefly 209
ภาพที่ 8.12 สร้างวีดีโอจากข้อความใน Firefly 209
ภาพที่ 8.13 Kodee AI สนับสนุนการตอบคำถามใน Hostinger 210

บทที่ 1
แนะนำหนังสือและ Generative AI

1.1 แนะนำหนังสือ

ผู้เขียนตัดสินใจอยู่ระยะหนึ่งว่าจะเขียนหนังสือดีหรือไม่ เหตุผลหลัก ๆ คือ การพัฒนา Generative AI มีการเปลี่ยนแปลงอยู่เรื่อย ๆ การถ่ายทอดจึงอาจจะยาก อย่างไรก็ตาม อาจจะต้องทำความเข้าใจว่าการพัฒนาระบบงานหรือโปรแกรมสมัยนี้เป็นการพัฒนาไป ใช้ไป รอรับ Feedback กลับมาจากผู้ใช้ เพื่อเปลี่ยนแปลงและปรับปรุง ไม่ใช่การพัฒนาแบบที่ทำให้เสร็จทีเดียวแล้วนำไปใช้ การมีข้อผิดพลาดจึงเกิดขึ้นได้ง่ายโดยเฉพาะการไม่เข้ากันของเวอร์ชัน ในเวอร์ชันที่ใหม่กว่าก็อาจจะไม่ได้ครอบคลุมบางส่วนของเวอร์ชันที่เก่ากว่า แต่ถ้านักพัฒนาตัดสินใจที่จะ Deprecate ก็หมายความว่าเลิกแนะนำให้ใช้แต่ก็ยังคงมีอยู่ สิ่งที่กระทบกับผู้ใช้งานอีกเรื่องหนึ่งก็คือ การย้ายหน้าจอ เมนู หรือฟังก์ชัน อาจสร้างความสับสนในการใช้งาน ดังนั้น การอธิบายหรือบรรยายเรื่องของระบบงาน อาจทำได้โดยการยึดหัวข้อหลัก ๆ ที่สำคัญในการพัฒนา และพยายามสร้างความเชื่อมโยงของระบบที่ออกแบบ

การเขียนหนังสือถ่ายทอดจึงอาจไม่ง่ายเท่ากับการทำวีดีโอคลิป แต่การเขียนหนังสือมีข้อดีกว่าที่ผู้อ่านสามารถมองเห็นความเกี่ยวข้องกันของสิ่งต่าง ๆ มองเห็นขอบเขตของความรู้ได้ดีกว่า และใช้เวลาน้อยกว่าการดูวีดีโอ หากมีสิ่งใดที่ไม่ได้อธิบายผู้อ่านก็สามารถไปค้นคว้าเพิ่มเติมจากสื่อประเภทอื่น ๆ ได้เพราะทราบหลักการพื้นฐานแล้วนั่นเอง นี่ก็คือเหตุผลที่ผู้เขียนตัดสินใจเขียนเพื่อถ่ายทอดให้คนทั่วไปได้รู้จัก Generative AI นั่นเอง

ขอบเขตในการอธิบายในหนังสือเล่มนี้ ได้แก่

(1) แนะนำความรู้เกี่ยวกับ Generative AI เน้นที่ LLMs และ การใช้งาน ChatGPT ในระดับพื้นฐาน
(2) เรียนรู้ ChatGPT Platform และ OpenAI Platform แบบเบื้องต้น และเข้าใจความสามารถในด้านต่าง ๆ ของแพลตฟอร์ม
(3) เรียนรู้ความสามารถของ ChatGPT กับการเขียนพรอมต์ แต่ไม่ได้อธิบาย Prompt Engineering อย่างไรก็ตาม การเขียนพรอมต์ที่อธิบายในเล่มนี้สามารถนำไปต่อยอดได้
(4) ทำความรู้จัก API กับการใช้งาน OpenAI API

(5) สร้างแอปพลิเคชันแบบไม่เขียนโปรแกรม และเขียนบ้างเล็กน้อยโดยแนะนำวิธีพัฒนาเบื้องต้น และการนำไปใช้

1.2 AI กับ Generative AI

AI คืออะไร ? เกี่ยวข้องอย่างไรกับ Generative AI และทำไม Generative AI ถึงมีความสำคัญ ก่อนอื่นต้องทำความรู้จัก AI ก่อน

ปัญญาประดิษฐ์ (Artificial Intelligence) เป็นความรู้แขนงหนึ่งซึ่งศึกษาเกี่ยวกับโครงสร้างและฟังก์ชันการทำงานของระบบชาญฉลาด (Intelligent systems) ซึ่งมีการทำงานด้วยระบบประสาทซึ่งมีตัวรับ (Receptor) และหน่วยปฏิบัติงาน (Affector) เชื่อมโยงเป็นเครือข่าย มีหน่วยความจำชั่วคราวและหน่วยความจำระยะยาว การศึกษาต่าง ๆ ในปัญญาประดิษฐ์จึงเป็นการศึกษาการทำงานของสมองและหน่วยรับรู้ของระบบชาญฉลาดปัญญาประดิษฐ์ โดยศึกษาไปที่การสะท้อนตอบต่อสิ่งที่มีเงื่อนไขและไม่มีเงื่อนไข การอนุมานหาข้อสรุป และการแสดงพฤติกรรมของระบบเมื่อเรียนรู้หรือได้รับการฝึกฝน อาจกล่าวกว้าง ๆ ว่าปัญญาประดิษฐ์เป็นศาสตร์ด้านหนึ่งในวิทยาการคอมพิวเตอร์ที่สอนให้เครื่องจักรหรือหมายถึงคอมพิวเตอร์สามารถทำงานได้เสมือนเป็นมนุษย์โดยมีสมองกลเป็นองค์ประกอบหลักในการทำงาน

งานที่สามารถประยุกต์ปัญญาประดิษฐ์ได้ก็คือ "งานที่มุ่งเน้นให้คอมพิวเตอร์สามารถทำงานได้แทนมนุษย์ เช่น การทำงานที่มีการทำซ้ำ ๆ และงานที่สามารถตัดสินใจได้โดยใช้เงื่อนไข"

การศึกษาปัญญาประดิษฐ์เกี่ยวข้องกับสาขาวิชาความรู้ในหลาย ๆ เรื่อง อาทิ

- ระบบผู้เชี่ยวชาญ (Expert System)
- การเรียนรู้ของเครื่องจักร (Machine Learning)
- การอนุมานด้วยตรรกะ (Logic reasoning)
- ระบบเอเจนต์ (Agent system)
- วิศวกรรมความรู้ (Knowledge engineering)

1.2.1 การเรียนรู้ของเครื่องจักร

ปัญญาประดิษฐ์ครอบคลุมหลาย ๆ เรื่องข้างต้น ทุก ๆ เรื่องนั้นต่างมีความเกี่ยวข้องกัน และสามารถประยุกต์การทำงานเข้าด้วยกัน แต่หลาย ๆ คนอาจรู้จักการเรียนรู้ของเครื่องจักรเป็นส่วนใหญ่ เพราะเป็นด้านที่มีการประยุกต์ใช้กับระบบงานต่าง ๆ อย่างกว้างขวางไม่ว่าจะเป็นด้านธุรกิจ การแพทย์ การประมวลผลภาพ การวิเคราะห์ข้อความ

การเรียนรู้ของเครื่องจักรอาจจำแนกออกเป็นสามประเภท ได้แก่

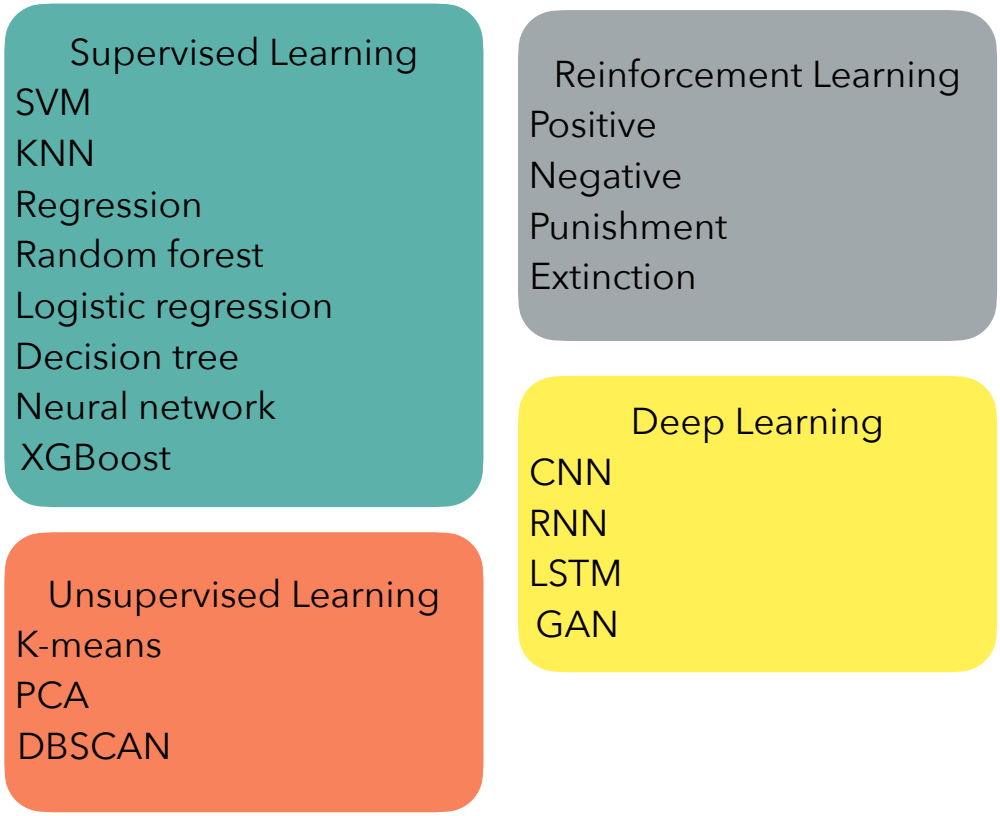
- การเรียนรู้ที่มีผู้สอน (Supervised Learning)
- การเรียนรู้ที่ไม่มีผู้สอน (Unsupervised Learning) และ
- การเรียนรู้แบบแรงเสริม (Reinforcement Learning)

กระบวนการเรียนรู้ของเครื่องจักรที่มีผู้สอนมีข้อมูลชุดเทรน (Training data) เพื่อใช้ในการสอน และชุดทดสอบ (Test data) เพื่อประเมินความแม่นยำของโมเดล การเทรนหรือฝึกฝนให้โมเดลทำงานได้เก่งจะต้องมีข้อมูลชุดเทรนเป็นจำนวนมาก อย่างไรก็ตาม ถึงแม้ว่าโมเดลจะเก่งขนาดไหนแต่เมื่อเจอข้อมูลชุดทดสอบที่ไม่รู้จักมาก่อนก็อาจจะทำให้ประสิทธิภาพในการทำงานไม่ดีนัก การเรียนรู้ที่ไม่มีผู้สอนใช้วิธีการวิเคราะห์จุดข้อมูลซึ่งก็คือตัวเลขทางคณิตศาสตร์เป็นหลัก การเรียนรู้แบบแรงเสริมมีทั้งการเสริมแรงทางบวก เช่น การให้รางวัล คำชมเชย หรือ ผลตอบแทน และการเสริมแรงทางลบ เช่น การทำโทษ

เทคนิคเหล่านี้ถูกนำไปใช้ในการจำแนก และการทำนายเป็นหลัก แต่ละเทคนิคล้วนมีข้อดีและข้อเสียแตกต่างกันไป อาจมีความเหมาะสมในการประยุกต์ใช้แก้ปัญหาในโดเมนหรือเรื่องที่ต่างกัน ในบางเทคนิคอาจสามารถเปรียบเทียบประสิทธิภาพระหว่างกันได้ แต่ในบางเทคนิคอาจมีวัตถุประสงค์ในการใช้งานซึ่งไม่สามารถนำมาพิจารณาเปรียบเทียบกันได้

ภาพที่ 1.1 การเรียนรู้ของเครื่องจักร

การสอนให้คอมพิวเตอร์เรียนรู้มีทั้งสอนให้สามารถจำแนกข้อมูล และทำนายข้อมูลที่ไม่เคยเห็นมาก่อน ซึ่งโมเดลหลัก ๆ มีสองประเภท คือ โมเดลถดถอย (Regression model) และโมเดลจำแนก (Classification model) แบบแรกใช้ในการทำนายค่าซึ่งเป็นตัวเลขจำนวนจริง เช่น การทำนายเปอร์เซ็นต์การเปลี่ยนแปลงของราคาหุ้นในวันพรุ่งนี้ หรือการทำนายราคาบ้าน ส่วนแบบหลังใช้เพื่อทำนายการจำแนก เช่น จำแนกว่าเป็นผู้ป่วยอาจเป็นมะเร็งหรือไม่เป็นมะเร็ง และการจำแนกภาพสัตว์ว่าเป็นสัตว์ประเภทใด (อาทิ ม้า ไก่ ช้าง) ภาพที่ 1.2 แสดงเทคนิคต่าง ๆ ในการเรียนรู้ของเครื่องจักร และการเรียนรู้เชิงลึก

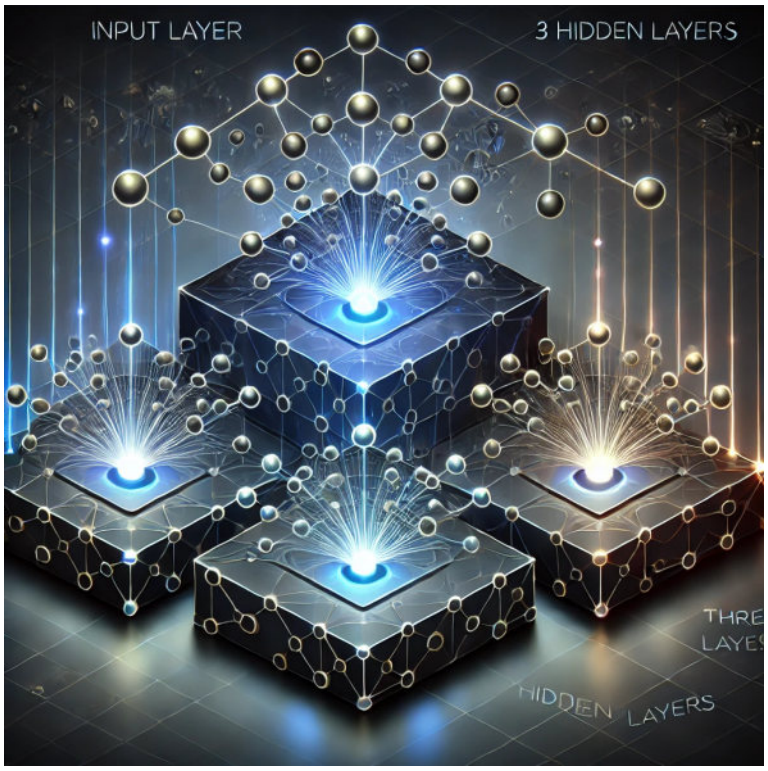


ภาพที่ 1.2 เทคนิคการเรียนรู้ด้วยเครื่องจักร

1.2.2 การเรียนรู้เชิงลึก

การเรียนรู้เชิงลึก (Deep Learning) เป็นวิวัฒนาการของเครือข่ายประสาทเทียมที่มีความซับซ้อนของโมเดลมากขึ้น อาทิ การกำหนดเลเยอร์ซ่อน (Hidden layer) มากขึ้น รวมถึงการปรับปรุงประสิทธิภาพด้วยการนำผลลัพธ์ที่เปลี่ยนผ่านไปในแต่ละเลเยอร์ย้อนเข้ามาเป็นอินพุตเพื่อทำให้โมเดลสามารถจำ และปรับปรุงประสิทธิภาพการเรียนรู้ให้ดีขึ้น ตัวอย่างโมเดลการเรียนรู้เชิงลึก ได้แก่ RNN

(Recurrent Neural Network) CNN (Convolution Nueral Network) ที่มีความนิยมในการพัฒนาแอปพลิเคชันซึ่งเกี่ยวข้องกับคอมพิวเตอร์วิทัศน์ (Computer Vision)



ภาพที่ 1.3 การเรียนรู้เชิงลึก

ปัญหาใหญ่ของการเรียนรู้ของเครื่องจักรรวมถึงการเรียนรู้เชิงลึก คือการเตรียมชุดข้อมูลเทรน และชุดข้อมูลทดสอบ โดยเฉพาะการเตรียมข้อมูลเทรนนั้นมีอุปสรรคค่อนข้างมาก อาทิ จำนวนข้อมูลที่อาจมีน้อยเกินไป และอาจต้องใช้เวลาในการเตรียมข้อมูลมาก เช่น การเก็บรวบรวมข้อมูลเทรนของผู้ป่วยที่เข้ามารักษาอาจต้องใช้เวลานาน การมีจำนวนข้อมูลที่น้อยเกินไปและไม่สมดุลก็นำไปสู่ความไม่แม่นยำในการทำนายของโมเดลได้ อาทิ หากข้อมูลของผู้ป่วยที่ไม่เป็นมะเร็งมีจำนวนมากกว่าจำนวนข้อมูลของผู้ป่วยที่เป็นมะเร็งก็จะเกิดอคติในการทำนายซึ่งมักจะมีแนวโน้มที่จะทำนายว่าไม่เป็นมะเร็งมากกว่าเป็นมะเร็ง

ด้วยเหตุนี้ Generative AI จึงเข้ามามีบทบาทเพื่อแก้ไขปัญหาดังกล่าวโดย Generative AI มีความสามารถในการสร้างชุดข้อมูลซึ่งข้อมูลที่ถูกสร้างจะเป็นข้อมูลที่จัดว่าคล้ายข้อมูลชุดเทรนที่มีอยู่เดิม การสร้างข้อมูลเพิ่มนี้ทำให้ได้ข้อมูลที่มีคุณลักษณะหลากหลายแต่จัดว่ามีความสอดคล้องกับข้อมูลชุดเดิม เมื่อจำนวนข้อมูล

มีมากขึ้นก็จะทำให้เกิดความสมดุลของข้อมูลและเป็นประโยชน์ในการทำนายได้ดี อย่างไรก็ตาม การจัดหาชุดข้อมูลเทรนนั้นยังอยู่บนความเชื่อที่ว่าข้อมูลมีความเชื่อถือได้ เช่น อาจผ่านการตรวจทานจากผู้เชี่ยวชาญในเรื่องนั้น ๆ มาก่อน

1.2.3 Generative AI

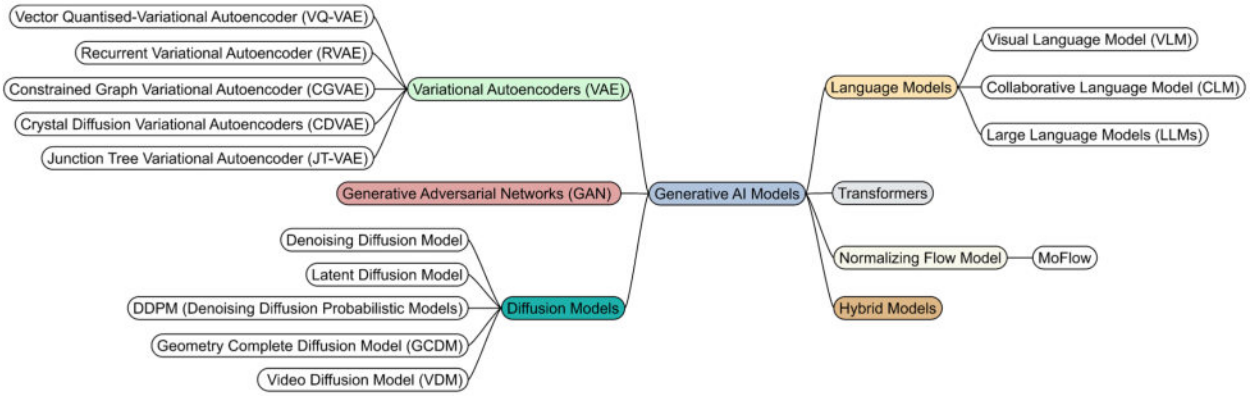
Generative AI เป็นโมเดลเรียนรู้จากข้อมูล และสร้างเนื้อหาใหม่ที่คล้ายกับข้อมูลต้นแบบ นอกจากการสร้างข้อมูลแล้วยังมีความสามารถในการแปลงข้อมูล เช่น จากข้อความเป็นภาพ จากเสียงเป็นข้อความ และยังมีความคิดสร้างสรรค์ มีการประยุกต์โมเดลต่าง ๆ ของการเรียนรู้ด้วยเครื่องจักรและการเรียนรู้เชิงลึก และสร้างโมเดลใหม่ที่มีความซับซ้อนมากขึ้น รวมถึงประยุกต์ศาสตร์อื่น ๆ อาทิ การสืบค้นข้อมูล (Information retrieval) การประมวลผลภาษาธรรมชาติ (Natural Language Processing) การอนุมานตรรกะ (Logical inference) การประมวลผลภาพ (Image processing) และการรู้จำ (Recognition) สิ่งเหล่านี้อาจสอดแทรกในการประมวลผลลำดับใดก็ได้ซึ่งผู้พัฒนากำหนด อีกทั้งการพัฒนา Generative AI ต้องใช้วิธีการประมวลผลข้อมูลขนาดใหญ่หรือที่รู้จักกันด้วยคำว่า BigData ซึ่งต้องใช้ระบบคอมพิวเตอร์จำนวนมากในการประมวลผล การเกิดขึ้นของ Generative AI ไม่สามารถสร้างได้ด้วยคนเพียงคนเดียวหรือระบบคอมพิวเตอร์หลักร้อย แต่ต้องมีความพร้อมในหลาย ๆ อย่าง เช่น ทีมนักพัฒนา ระบบคอมพิวเตอร์ ข้อมูล และเงินจำนวนมากที่ใช้ในการลงทุน หลาย ๆ บทความกล่าวว่าโมเดลของ Generative AI นั้นมีพารามิเตอร์มากเป็นล้านพารามิเตอร์ แต่ละพารามิเตอร์จะมีค่าน้ำหนักแสดงความสำคัญในแต่ละบริบทที่กำลังถูกประมวลผลในโมเดล

มาถึงตรงนี้ก็อาจจะสรุปได้ว่า Generative AI มีความซับซ้อนมาก ๆ และคงไม่ง่ายเหมือนการเขียนโปรแกรมเรียกฟังก์ชันส่งพารามิเตอร์เพียงไม่กี่ตัว การพัฒนาที่ก่อกำเนิด Generative AI มีวิวัฒนาการมาจากการพัฒนาปัญญาประดิษฐ์ที่ถูกเรียนรู้ พัฒนา และวิจัยมามากกว่า 80 ปี แต่วิทยาการด้านนี้เพิ่งมาถึงจุดที่ทุกคนในโลกเกิดความเชื่อถือ และต้องการใช้หรือเข้าถึงข้อมูล

ภาพที่ 1.4 Generative AI

ระบบอันชาญฉลาดซึ่งเป็น Generative AI มีหลายระบบ เช่น ChatGPT (OpenAI), Claude AI (Anthropic), Gemini (Google), LLaMa (Facebook), Grok (xAI ของ Elon Musk) และ DeepSeek

Generative AI อาจเป็นระบบที่เชี่ยวชาญสำหรับงานเฉพาะด้านก็ได้ เช่น ภาพ เสียง หรือวีดีโอเพียงอย่างเดียว กลไกการทำงานของแต่ละระบบขึ้นกับลักษณะของโมเดลที่พัฒนา



ภาพที่ 1.5 การจำแนกโมเดล Generative AI [1]

ChatGPT เป็น Generative AI ในกลุ่ม Language Models (ภาพที่ 1.5) พัฒนาโดยบริษัท OpenAI เป็น Generative AI ที่ถูกกล่าวขานกันในปัจจุบัน และเป็นที่ประจักษ์ว่าเป็นระบบงานที่ชาญฉลาดแม้ว่าข้อมูลอาจจะยังไม่ถูกต้องร้อยเปอร์เซ็นต์ ซึ่งมีงานวิจัยหลาย ๆ งานได้ประเมินผลมาแล้ว (เช่น [2] และ [3])

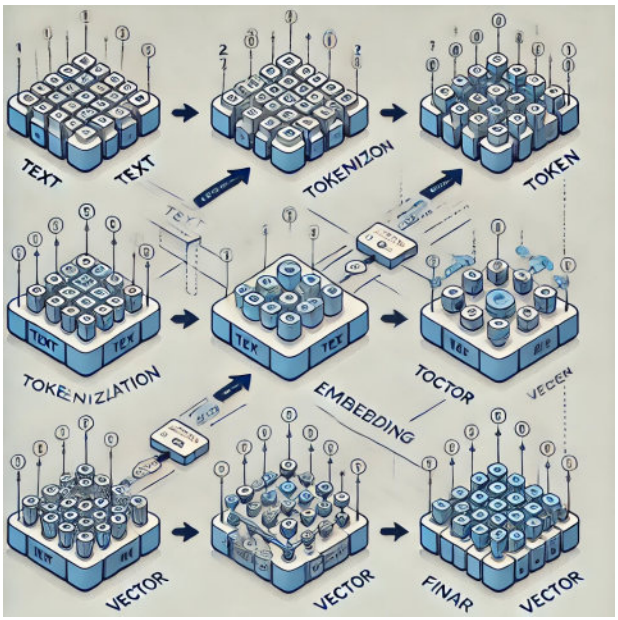
ผู้เขียนเชื่อว่าการพัฒนา Generative AI จะทำให้ชีวิตของมนุษย์ง่ายขึ้นในด้านการเสาะแสวงหาข้อมูลและความรู้ และสามารถนำข้อมูลเหล่านั้นมาใช้ในการพัฒนาอาชีพตนเองต่อไป อาจเปรียบเปรยได้ว่าก็เหมือนกับการใช้ Google ค้นหาข้อมูลในยี่สิบสามสิบปีที่ผ่านมา แต่ ณ วันนี้ Generative AI ทำให้โลกของข้อมูลและระบบงานหรือแอปพลิเคชันสามารถประสานเข้าด้วยกันอย่างง่ายดาย คงไม่ต้องห่วงว่าเขาจะมาแย่งงานเรา แต่อาจจะต้องเป็นห่วงว่าจะหาเงินซื้อเทคโนโลยี AI เหล่านี้ได้อย่างไรบ้าง อย่างไรก็ตาม ผู้เขียนเชื่อว่าเทคโนโลยีจะราคาถูกลงตามกาลเวลาเหมือนเครื่องคอมพิวเตอร์ในอดีตที่มีราคาแสนแพง แต่ถูกลงอย่างไม่น่าเชื่อในเวลาต่อมา อาจมีเพียงการปรับเปลี่ยนวิถีชีวิตในการทำงานที่จะต้องเปลี่ยนไปโดยที่มนุษย์ทุกคนควรจะเรียนรู้เพื่อให้สามารถนำประโยชน์ของ Generative AI มาใช้ได้ในชีวิตประจำวัน

1.3 LLMs

1.3.1 LLMs กับการประมวลผลข้อความ

LLMs เป็นคำย่อของ Large Language Models ในการทำงานของ ChatGPT ต้องพึ่งพา LLMs ดังนั้นจะต้องเข้าใจการทำงาน และรู้จัก LLMs สักเล็กน้อย LLMs จัดการคำที่ถูกแบ่งออกมาจากข้อความในชุดเทรนด้วยกรรมวิธีที่เรียกว่า Tokenization ภาษาธรรมชาติหนึ่ง ๆ ก็จะต้องใช้พาร์สเซอร์ (Parser) เฉพาะสำหรับภาษานั้นในการประมวลผลข้อความและโทเค็น สิ่งหนึ่งที่อาจเป็นผลกระทบจากการแบ่งคำก็คือความแม่นยำในการแบ่งคำ และแน่นอนว่าจะส่งผลต่อค่าใช้จ่ายที่เราใช้งานกับ โมเดล GPT ซึ่งเป็น LLMs ของ ChatGPT

ภาพที่ 1.6 แสดงการประมวลผลข้อความ จากข้อความไปเป็นเว็กเตอร์ ซึ่งมีรายละเอียดดังนี้



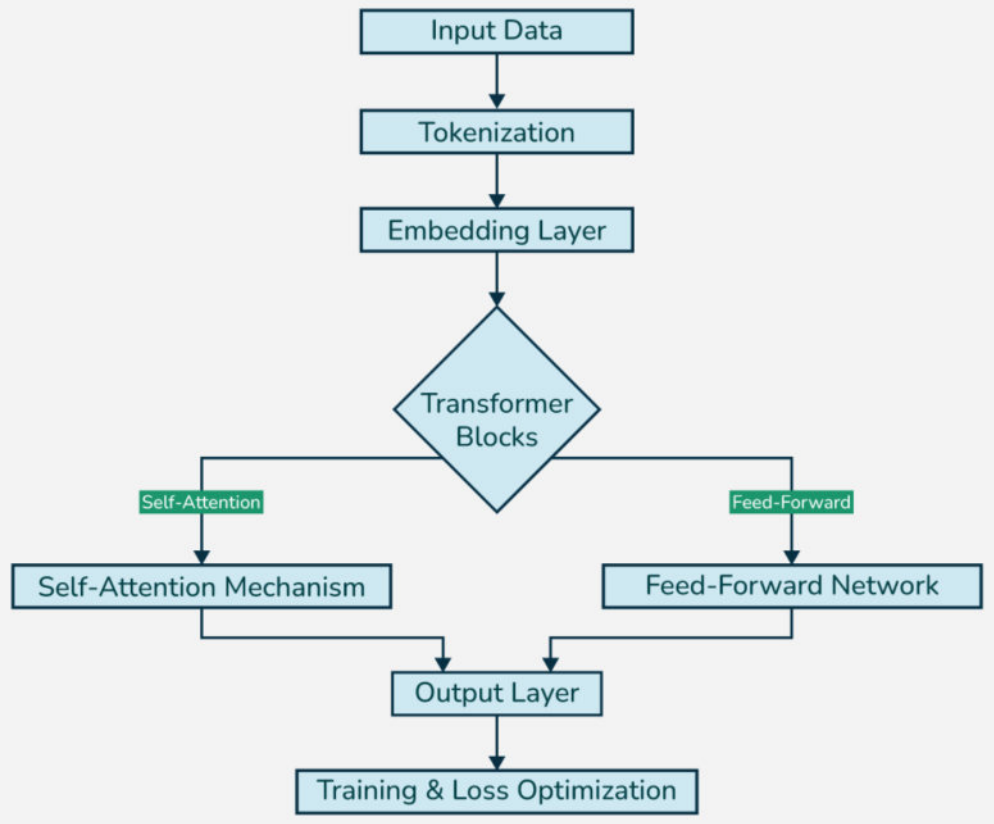
ภาพที่ 1.6 Text, Tokenization, Token, Embedding, Vector

- เท็กซ์ (Text) คือข้อความ อาจเป็นประโยค หรือเป็นข้อความของเอกสาร เป็นข้อมูลนำเข้าสู่โมเดล
- การแบ่งเป็นโทเค็น (Tokenization) คือกระบวนการแบ่งข้อความออกเป็นหน่วยย่อยที่เรียกว่าโทเค็น โทเค็นอาจเป็นคำ ส่วนของคำ หรือตัวอักษรก็ได้ ตัวอย่างเช่น ประโยค "I love NLP." อาจถูกแบ่งเป็นโทเค็น "I", "love", "NLP" และ "."
- โทเค็น (Token) คืออักขระหรือกลุ่มอักขระ
- การฝังตัวหรือเอมเบดดิง (Embedding) โทเค็นจะถูกแปลงหรือแทนด้วยหมายเลข เช่น "I", "love", "NLP" และ "." อาจแทนด้วย 1,2,3 และ 4
- เวกเตอร์ (Vector) คือรูปแบบแสดงการฝังตัวของโทเค็นที่มีหลายมิติการแสดงข้อความในเวกเตอร์ทำให้ง่ายต่อการประมวลผลและจัดเก็บข้อมูลในคอมพิวเตอร์

ลำดับขั้นตอนการทำงานข้างต้นทำให้โมเดล LLMs สามารถแปลงภาษามนุษย์ให้อยู่ในรูปแบบตัวเลขที่สามารถนำไปวิเคราะห์ และสร้างข้อความขึ้นมาใหม่ได้ ส่งเสริมการใช้งานที่หลากหลายในการประมวลผลภาษาธรรมชาติ

1.3.2 สถาปัตยกรรมของ LLMs

LLM (Large Language Model) เป็นโมเดลปัญญาประดิษฐ์ที่ใช้ในการประมวลผลและสร้างข้อความที่ซับซ้อน โดยมีพื้นฐานมาจาก Transformer Architecture ซึ่งพัฒนาโดยทีมของ Google ในค.ศ. 2017 โมเดลสามารถเข้าใจ และสร้างข้อความได้อย่างแม่นยำโดยใช้เทคนิคที่เรียกว่า Self-Attention และการเรียนรู้เชิงลึก (ภาพที่ 1.7)



ภาพที่ 1.7 สถาปัตยกรรม LLM [4]

สถาปัตยกรรมของ LLMs ทำการแปลงข้อความจากเว็กเตอร์เอมเบดดิงแล้วส่งให้กับอัลกอริทึม Self-Attention และเครือข่าย Feed-forward แล้วประมวลผลสู่ Output layer

Self-Attention ทำให้โมเดลโฟกัสไปที่คำสำคัญในประโยค ตัวอย่างเช่น ประโยค "The cat sat on the mat." โมเดลจะทำนายความสัมพันธ์ระหว่าง "cat" กับ "sat" และ "mat" กับ "sat" ดังนั้นจึงสร้างประโยคดังกล่าวได้ ตำแหน่งของโทเค็นพิจารณาจากเลเยอร์ Positional Encoding ดังนั้น "sat" มีลำดับมาก่อน "mat"

Feed-Forward Network มีหน้าที่ช่วยแปลงข้อมูลและเพิ่มความซับซ้อนของการประมวลผลข้อความ ในส่วนนี้ทำให้เกิดการสร้างข้อความที่หลากหลายได้

Output Layer ทำหน้าที่ในการสร้างคำตอบของโมเดล ตัวอย่างเช่น อินพุต คือ ["ChatGPT", "is"] โมเดลทำนายคำถัดไปจากค่าความน่าจะเป็น ["amazing" (0.75), "powerful" (0.80), "great" (0.60)] หากเวกเตอร์แสดงความใกล้เคียงของความหมายของคำ ดังนั้น ประโยคที่จะสร้างได้ คือ "ChatGPT is powerful" ในเลเยอร์นี้ยังเกี่ยวข้องกับการแสดงผลลัพธ์ในรูปแบบ (เช่น โค้ด ตาราง) ต่าง ๆ

Training & Loss optimization layer เป็นเลเยอร์ปรับปรุงการเรียนรู้ของโมเดล ซึ่งการเรียนรู้เชิงลึกด้วยเครือข่ายประสาทเทียมหลายชั้นถูกนำเข้ามาใช้ในการเรียนรู้ และปรับปรุงค่าของเวกเตอร์ให้อยู่ในช่วงที่เหมาะสม การปรับค่าพารามิเตอร์ในเลเยอร์นี้ก็จะทำให้ได้ผลลัพธ์ที่แตกต่างกันใน Output Layer ด้วย

1.3.3 เอมเบดดิง

เอมเบดดิง (Embedding) คือกระบวนการแปลงข้อความหรือเท็กซ์ให้เป็นตัวเลข จากกระบวนการทำงานของ LLMs ข้อความหรือเท็กซ์ที่สกัดได้ (โทเค็น) จะถูกนำไปแสดงในเวกเตอร์ ตัวเลขในเวกเตอร์อาจนำไปใช้ประมวลผลในเรื่องต่าง ๆ เช่น ความคล้ายเชิงความหมาย (Semantic similarity) และการปรากฏพร้อมกัน (Co-occurence) ดังนั้นจึงสามารถนำเวกเตอร์มาประมวลผลเพื่อพัฒนาแอปพลิเคชันต่าง ๆ เช่น ระบบแนะนำ ระบบค้นหาข้อมูล

ตัวอย่างเอมเบดดิง สมมติว่ามีข้อความอยู่ 3 ประโยคต่อไปนี้

(1)"Apple is a fruit"

(2)"Apple makes iPhones"

(3)"Banana is yellow"

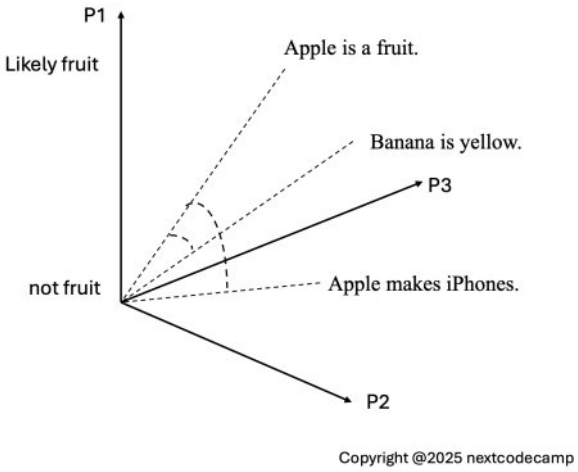
สามารถสกัดคำ (โทเค็น) ออกมาได้ 6 คำ เช่น Apple (เป็นผลไม้), fruit (ผลไม้), Apple (บริษัท), iPhones (โทรศัพท์), Banana (ผลไม้), และ yellow (สี)

LLMs อาจวิเคราะห์ความเกี่ยวข้องของคำในแง่มุมต่าง ๆ เช่น

- ในประโยคที่ 1 และ 2 นั้น "Apple" มีความหมายต่างกัน ในแง่มุมคือมีความคล้ายในการเขียนแต่ความหมายต่าง อาจจัดว่าเกี่ยวข้องกันน้อยมาก แต่เขียนเหมือนกัน
- ในประโยคที่ 1 และ 3 นั้น "Apple" กับ "Banana" ต่างก็เป็นผลไม้ ดังนั้นโทเค็นทั้งสองใกล้เคียงกันมากกว่าข้อแรก

- ในประโยคที่ 2 และ 3 นั้น "Apple" กับ "Banana" มีความเกี่ยวข้องกันน้อยมาก

LLMs จะคำนวณ Cosine similarity เพื่อหาคะแนนความคล้ายของประโยค ซึ่งคำนวณจากคำ ภาพที่ 1.8 แสดงตัวอย่าง Cosine similarity มุมแสดงระยะห่างของประโยคนั่นเอง



ภาพที่ 1.8 Cosine similarity

OpenAI สนับสนุนการทำงานกับเวกเตอร์เอมเบดดิง โค้ดที่ 1.1 แสดงตัวอย่างการคำนวณหาค่าในเวกเตอร์ของประโยค "Your text string goes here" ผลลัพธ์แสดงเวกเตอร์จำนวน 1,536 มิติ การใช้โมเดลที่แตกต่างออกไปก็จะมีจำนวนมิติที่ต่างกันไป

โค้ดที่ 1.1 ตัวอย่างการทำงานกับ Embedding API

```
import openai



# 1. Set your OpenAI API key

OPENAI_API_KEY = "Your-API-Key"
```

```
# 2. Initialize OpenAI client

client = openai.OpenAI(api_key=OPENAI_API_KEY)



# Get OpenAI embeddings for text

response = client.embeddings.create(

    input="Your text string goes here",

    model="text-embedding-3-small"

)



print(response.data[0].embedding)
```

ผลลัพธ์จะแสดงเวกเตอร์ขนาดใหญ่ซึ่งมีถึง 1,536 มิติ ลองจินตนาการกราฟที่มีแกน 1,536

```
[0.005172153003513813, 0.017217181622982025, -0.018686940893530846, -0.01854696311056
0.03026304580271244, 0.027659472078084946, 0.003663900075480342, 0.011233161203563213
6980969812721014, 0.01585940271615982, -0.0012702919775620103, -0.007873711176216602,
76187777519, -0.02751949429512024, 0.00991037767380476, -0.040397386997938156, 0.0499
51745093, 0.0302350502461195, -0.013717753812670708, 0.03295060619711876, 0.017287170
4, -0.0017374654999002814, 0.02042265608906746, 0.040789321064949036, -0.037737820297
-0.05002781003713608, 0.0241740420460701, -0.0551229752600193, -0.03227871656417847,
0646791458, 0.01469759363681078, -0.01566343568265438, -0.04132123664021492, 0.022200
186, 0.044960640370845795, 0.007107336539775133, -0.024118050932884216, 0.05240742117
-0.032222725450992584, 0.01626533642411232, 0.046444397419691086, 0.02382409945130348
05371620412915945, 0.04395280405879021, -0.006148493383079767, 0.007306803949177265,
```

จากภาพข้างต้นตัวเลขที่แสดงเป็นเพียงส่วนเล็ก ๆ ในเวกเตอร์ เวกเตอร์ที่ได้สามารถนำไปคำนวณหาค่าความคล้าย เช่น ใช้ Cosine similarity คำนวณ ตัวอย่างโค้ด Python คำนวณ Similarity ของประโยคอาจดูในตัวอย่างแอปพลิเคชันในหัวข้อ 7.8

1.3.4 การสร้างประโยคแบบ Nucleaus sampling

จากที่เคยเกริ่นว่า Generative AI มีความสามารถในการสร้างประโยคที่มีความคิดสร้างสรรค์ เช่น การใช้คำที่หลากหลายมากขึ้น วิธีการสร้างประโยคแบบ Nucleaus sampling ทำให้ ChatGPT สามารถสร้างสรรค์ข้อความใหม่ ๆ ได้ สมมติว่ามี 6 ประโยคที่เกี่ยวข้องกับประโยค 'Generative AI Book' แต่ละประโยคมีค่าความเป็นไปได้เรียงลำดับจากมากไปหาน้อยดังนี้

- OpenAI Platform (Prob = 0.4)
- Coding by Python (Prob = 0.3)
- Create application for Generative AI (Prob = 0.2)
- From application to powerful application (Prob = 0.1)
- DALL·E AI Image creator (Prob = 0.07)
- Whisper audio to text (Prob = 0.04)

ในโมเดล GPT หากกำหนดพารามิเตอร์ Top-p = 0.8 นั่นหมายความว่าการสร้างประโยคถัดจาก 'Generative AI Book' จะต้องเลือกประโยคที่รวมกันแล้วได้ 0.8 เป็นอย่างต่ำ เช่น นำประโยค 1 ถึง 3 มาสร้างเป็นประโยคถัดไป ดังนั้นจะได้ประโยคยาวขึ้นคือ 'Generative AI Book OpenAI Platform Coding by Python Create application for Generative AI' อนึ่ง ค่าคะแนนของประโยคหลาย ๆ ประโยคอาจเท่ากันก็ได้ ดังนั้นถ้ากำหนดระดับ Top-p ยิ่งสูงก็ยิ่งจะสร้างได้หลายประโยคที่หลากหลาย!

พารามิเตอร์ Top-p มีความใกล้เคียงกับพารามิเตอร์ Temperature แต่ต่างกันตรงที่ Temperature ให้ความสำคัญกับการสุ่มคำ/โทเค็นถัดไป ในขณะที่ Top-p กำหนดข้อจำกัดในการเลือกโทเคนเพื่อสร้างประโยคนั่นเอง ตัวอย่างศึกษาการกำหนดค่าพารามิเตอร์ที่มีผลต่อการสร้างประโยคอธิบายไว้ในหัวข้อที่ 4.6