

การศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรม
โทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
STUDY AND DEVELOPMENT OF RECOMMENDATION SYSTEM ABOUT
TELECOMMUNICATION ENGINEERING, KMITL



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมโทรคมนาคม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรม
โทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
STUDY AND DEVELOPMENT OF RECOMMENDATION SYSTEM ABOUT
TELECOMMUNICATION ENGINEERING, KMITL

โดย

นางสาวชญญา	ปรีชา	62010181
นายธรรมรักษ์	แก้วคำบัง	62010405
นางสาวพัชรกิตติ์	ยางเดี่ยว	62010611

อาจารย์ที่ปรึกษา

ผศ.ดร.ธเนศ พัฒนธาดาพงษ์

ผศ.ดร.สมเกียรติ ฤกษ์วัลย์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2565

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2565

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรมโทรคมนาคม สถาบัน
เทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

STUDY AND DEVELOPMENT OF RECOMMENDATION SYSTEM ABOUT
TELECOMMUNICATION ENGINEERING, KMITL

ผู้จัดทำ

- | | | |
|---------------------|------------|----------|
| 1. นางสาวชญญา | ปรีชา | 62010181 |
| 2. นายธรรมรักษ์ | แก้วคำบั้ง | 62010405 |
| 3. นางสาวพัชรกิตติ์ | ยางเดี่ยว | 62010611 |



..... อาจารย์ที่ปรึกษา

(ผศ.ดร.ธเนศ พัฒนธาดาพงษ์)



..... อาจารย์ที่ปรึกษาร่วม

(ผศ.ดร.สมเกียรติ ฤกษ์วีระบุญ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

การดำเนินปฏิญานิพนธ์เรื่อง “การศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง” จะไม่สำเร็จล่วงไปได้ด้วยดี หากไม่ได้รับความช่วยเหลือ ความอนุเคราะห์และการสนับสนุนอย่างดียิ่งจาก ผศ.ดร.ธเนศ พัฒนาธาดาพงษ์ และ ผศ.ดร.สมเกียรติ ฤกษ์วีระบุญ ที่ให้คำแนะนำ คำปรึกษาและแนวทางการแก้ไขปัญหาที่เป็นประโยชน์ต่อการศึกษา รวมถึงสนับสนุนสถานที่ และอุปกรณ์ต่างๆที่ใช้ระหว่างการจัดทำปฏิญานิพนธ์

ขอขอบคุณคณาจารย์ประจำภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังทุกท่าน ที่ได้อบรมสั่งสอน ประสิทธิ์ประสาทวิชาความรู้ และประสบการณ์ให้แก่ผู้จัดทำ

ขอขอบคุณผู้มีส่วนเกี่ยวข้องทุกท่านเป็นอย่างสูงที่คอยช่วยเหลือและกราบขอบพระคุณบิดา มารดา และครอบครัว ที่ให้ความรัก ความห่วงใย และเป็นกำลังใจที่สำคัญเสมอมาและที่สำคัญคือสนับสนุนให้โอกาสทางด้านการศึกษามีค่ายิ่งแก่ผู้จัดทำ

นางสาวชญญา ปรีชา
นายธรรมรักษ์ แก้วคำบั้ง
นางสาวพัชรกิติ์ ยางเดี่ยว
ผู้จัดทำ

การศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชา
วิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้า
คุณทหารลาดกระบัง
STUDY AND DEVELOPMENT OF RECOMMENDATION
SYSTEM ABOUT TELECOMMUNICATION
ENGINEERING, KMITL

โดย นางสาวชญญา ปรีชา 62010181
นายธรรมรักษ์ แก้วคำบัง 62010405
นางสาวพัชรกิติ์ ยางเดี่ยว 62010611

อาจารย์ที่ปรึกษา ผศ.ดร.ธเนศ พัฒนธาดาพงษ์
อาจารย์ที่ปรึกษาร่วม ผศ.ดร.สมเกียรติ ฤกษ์วัลย์

บทคัดย่อ

ปฏิญานิพนธ์นี้นำเสนอเกี่ยวกับการศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง เนื่องจากผู้ใช้งานมีความยุ่งยากที่ต้องหาข้อมูลจากเล่มหลักสูตรที่มีจำนวนมาก ปฏิญานิพนธ์นี้จึงมีวัตถุประสงค์เพื่ออำนวยความสะดวกให้แก่ผู้ใช้งาน การจัดเตรียมปฏิญานิพนธ์นี้ได้ทำการออกแบบโมเดลโดยใช้วิธีการของการประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) และใช้สถาปัตยกรรมของ Transformer หลังจากนั้นจึงทำการสร้างส่วนขยายเพื่อนำไปใช้งานบนเว็บไซต์และสามารถโต้ตอบผ่านข้อความหรือเสียงโดยแสดงผลเป็น Motion Graphic

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

This thesis present about the study and development of recommendation system about Department of Telecommunication Engineering, KMITL. Because users have difficulty of finding information from several curriculum books. So, the objective of this thesis is to make it more convenience for the users. In this thesis, the model was design by using the methods of Natural Language Processing (NLP) and architecture of transformer, then create the extension for use on website and interact through text or sound that it displayed as motion graphic.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
กิตติกรรมประกาศ	I
บทคัดย่อ	II
สารบัญ	IV
สารบัญรูป	X
บทที่ 1	1
บทนำ	1
1.1 ความเป็นมาและความสำคัญของปัญหา	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของโครงการ	1
บทที่ 2	2
ทฤษฎีและหลักการที่เกี่ยวข้อง	2
2.1 ระบบแนะนำข้อมูล	2
2.2 การประมวลผลภาษาธรรมชาติ (NLP)	3
2.3 TRANSFORMERS ARCHITECTURE	3
2.3.1 ATTENTION MODULE	7
2.4 Bidirectional Encoder Representations from Transformers: BERT	12
2.4.1 ATTENTION MODULE	12
2.4.2 FINE-TUNING BERT	13
2.5 Open-Domain Question Answering: ODQA	13
2.5.1 Open-Book QA: Retriever-Reader	14
2.6 Retrievers for Question-Answering System	15
2.7 Reader Model for Question-Answering System	17
2.7.1 กระบวนการเตรียมข้อมูล (Pre-processing)	18

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.8 การวัดประสิทธิภาพของโมเดล	21
2.8.1 Exact Match	21
2.8.2 F1 - Score	22
2.9 ฐานข้อมูล (KNOWLEDGE BASE)	23
2.9.1 ระบบจัดการฐานข้อมูล MYSQL	24
2.9.2 ภาษา STRUCTURE QUERY LANGUAGE (SQL)	24
2.10 MICROSERVICE	24
2.11 API (APPLICATION PROGRAMMING INTERFACE)	25
2.11.1 REST API (RESTFUL WEB SERVICES)	26
2.11.2 FASTAPI และ UVICORN	26
2.12 โพรโทคอล HTTP และ HTTPS	26
2.12.1 โพรโทคอล HTTP	26
2.12.2 โพรโทคอล HTTPS	27
2.13 เว็บไซต์ (WEBSITE)	27
2.13.1 WEB BROWSER	28
2.13.2 WEB SERVER	30
2.14 TEXT-TO-SPEECH	32
2.15 SPEECH-TO-TEXT	33
2.16 กราฟิกเคลื่อนไหว (MOTION GRAPHIC)	33
2.17 DOCKER	34
2.17.1 องค์ประกอบพื้นฐานของ DOCKER	35
2.17.2 ข้อแตกต่างของ DOCKER เมื่อเทียบกับ VIRTUAL MACHINE	35
2.18 GOOGLE CLOUD PLATFORM (GCP)	36
2.19 HEROKU	37

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.20 GIT	37
บทที่ 3	
การออกแบบและการจัดทำปฏิญญานิพนธ์	38
3.1 การออกแบบ	38
3.1.1 การออกแบบระบบรวม	38
3.1.2 การออกแบบและสร้าง READER MODEL	40
3.1.3 การออกแบบและสร้าง KNOWLEDGE BASE	53
3.1.4 การสร้างและออกแบบระบบของ RECOMMENDATION SYSTEM API	56
3.1.5 การ DEPLOY RECOMMENDATION SYSTEM บน GOOGLE CLOUD	60
3.1.6 การออกแบบกราฟิกเคลื่อนไหว (MOTION GRAPHIC)	61
3.1.7 การออกแบบหน้าเว็บไซต์สำหรับใช้งาน RECOMMENDATION SYSTEM	66
3.1.8 การออกแบบเว็บเซิร์ฟเวอร์	75
3.1.9 การสื่อสารระหว่างฝั่งผู้ใช้งานกับระบบหลังบ้านทั้งหมด (BACKE END)	85
3.1.10 การ DEPLOY เว็บเซิร์ฟเวอร์ไปบน HEROKU	87
3.2 เครื่องมือที่ใช้ในการทดลอง	89
3.2.1 GOOGLE COLAB PRO	89
3.2.2 VISUAL STUDIO CODE	89
3.2.3 JUPYTER NOTEBOOK	90
3.2.4 โปรแกรม POSTMAN	90
3.2.5 NODE.JS	90

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
3.2.6 GOOGLE CLOUD PLATFORM	90
3.2.7 DOCKER	90
3.2.8 MYSQL	90
3.2.9 ADOBE PREMIERE PRO	91
3.2.10 CLIP-STUDIO PAINT	91
3.3 การจัดเก็บผลการทดลอง	91
3.3.1 ทดสอบและประเมินประสิทธิภาพของ READER MODEL	91
3.3.2 ทดสอบการใช้งาน KNOWLEDGE BASE	91
3.3.3 ทดสอบระบบ RECOMMENDATION SYSTEM API	92
3.3.4 ทดสอบการสื่อสารระหว่างเว็บเซิร์ฟเวอร์ กับ API	92
3.3.5 ทดสอบการสื่อสารระหว่างฝั่งผู้ใช้งาน (CLIENT) กับระบบ หลังบ้านทั้งหมด (BACK-END)	92
3.3.6 ทดสอบการใช้งานเว็บไซต์ที่ขนาดหน้าจอแตกต่างกัน (RESPONSIVE WEB DESIGN)	92
3.3.7 ทดสอบการใช้งานระบบรวม	
บทที่ 4 ผลการทดลอง	93
4.1 ทดสอบและประเมินประสิทธิภาพของ READER MODEL	93
4.1.1 ทดสอบกระบวนการเตรียมข้อมูล (PRE-PROCESSING)	93
4.1.2 ทดสอบกระบวนการหาพารามิเตอร์ที่ดีที่สุด (HYPERPARAMETER OPTIMIZATION)	100
4.1.3 กระบวนการประเมินประสิทธิภาพของแบบจำลองด้วย ผลคะแนน EXACT MATCH และ F1	100

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
4.2 ทดสอบการใช้งาน KNOWLEDGE BASE	100
4.2.1 ทดสอบการเรียกดูข้อมูลบริบทจากตารางสำหรับเก็บข้อมูลบริบท	100
4.2.2 ทดสอบการเพิ่มคำถามไปที่ตารางสำหรับคำถามที่ระบบไม่สามารถตอบได้	101
4.3 ทดสอบระบบ RECOMMENDATION SYSTEM API	102
4.3.1 ทดสอบระบบตรวจสอบการตกทาย	102
4.3.2 ทดสอบระบบเตรียมข้อมูลคำถาม (QUESTION PREPROCESSING)	103
4.3.3 ทดสอบระบบเตรียมข้อมูลบริบท (CONTEXTS PREPROCESSING)	103
4.3.4 ทดสอบระบบค้นคืนข้อมูล (RETRIEVER)	103
4.3.5 ทดสอบการใช้งาน READER MODEL	104
4.3.6 ทดสอบการเชื่อมต่อ RECOMMENDATION SYSTEM API ด้วยโปรแกรม POSTMAN	104
4.4 ทดสอบการสื่อสารระหว่าง WEB SERVER กับ API	106
4.4.1 ทดสอบการส่ง-รับข้อมูลที่ WEB SERVER กับ RECOMMENDATION SYSTEM API	106
4.4.2 ทดสอบการส่ง-รับข้อมูลที่เว็บเซิร์ฟเวอร์กับ GOOGLE CLOUD TEXT-TO-SPEECH API	107
4.5 ทดสอบการสื่อสารระหว่างฝั่งผู้ใช้งานกับเว็บเซิร์ฟเวอร์	107
4.5.1 ทดสอบการร้องขอของหน้าเว็บไซต์ด้วยเว็บเบราว์เซอร์	107
4.5.2 ทดสอบการส่ง-รับข้อมูลระหว่างฝั่งผู้ใช้งานกับเว็บเซิร์ฟเวอร์	108
4.6 ทดสอบการใช้งานเว็บไซต์ที่ขนาดหน้าจอต่างกัน	108

สารบัญ (ต่อ)

	หน้า
4.7 ทดสอบการใช้งานระบบรวม	112
บทที่ 5	
สรุปผลและข้อเสนอแนะ	
5.1 สรุปผล	117
5.2 ข้อเสนอแนะ	118
บรรณานุกรม	119
ภาคผนวก ก กระบวนการ PRE-PROCESSING	128
ภาคผนวก ข READER MODEL	133
ภาคผนวก ค โปรแกรมของระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรมโทรคมนาคม	135
ภาคผนวก ง โปรแกรมสำหรับการสร้าง WEB SERVICE API	147
ภาคผนวก จ โปรแกรมที่ใช้ในการสร้าง CONTAINER IMAGE (DOCKERFILE)	149
ภาคผนวก ฉ โปรแกรมสำหรับหน้าเว็บไซต์	151
ภาคผนวก ช โปรแกรมสำหรับเว็บเซิร์ฟเวอร์	174

สารบัญรูป

รูปที่	หน้า	
2.1	สถาปัตยกรรมทั่วไปของ TRANSFORMERS	4
2.2	ส่วนประกอบของส่วนการเข้ารหัส	4
2.3	ส่วนประกอบของส่วนการถอดรหัส	7
2.4	แผนภาพ SCALED DOT-PRODUCT ATTENTION	9
2.5	แผนภาพ MULTI-HEAD ATTENTION	10
2.6	SELF-ATTENTION	11
2.7	MASKED SELF-ATTENTION	11
2.8	ENCODER-DECODER ATTENTION	11
2.9	LANGUAGE MODEL แบบดั้งเดิม (ซ้าย) เปรียบเทียบกับ MASKED LANGUAGE MODEL (ขวา)	12
2.10	ภาพรวมการทำงานของ OPEN-DOMAIN QUESTION ANSWERING (ODQA) ทั้ง 3 แบบ	14
2.11	โครงสร้างของระบบ OPEN-BOOK QA: RETRIEVER-READER	15
2.12	“INPUT_IDS” ที่ถูกถอดรหัส โดยสีเทาแสดงคำถาม สีฟ้าแสดงบริบท และสีน้ำเงินแสดงคำตอบ	17
2.13	โครงสร้างการทำงานของ READER MODEL	18
2.14	คู่คำถามและบริบทก่อนและหลังผ่านกระบวนการเตรียมข้อมูล	18
2.15	DICTIONARY KEY ของเอาต์พุตหลังผ่านกระบวนการเตรียมข้อมูล	19
2.16	ค่า INPUT_IDS ของแต่ละคำในประโยค	20
2.17	การแปลงกลับจาก INPUT_IDS ให้อยู่ในรูปแบบของประโยคที่ถูกโทเคน	20
2.18	ตัวอย่างการตัดคำโดยใช้อัลกอริทึม SENTENCEPIECE เพื่อรักษาช่องว่าง	21
2.19	ตาราง CONFUSION MATRIX	22
2.20	แสดงการทำงานของบริการเว็บไซต์ (WEBSITE)	28
2.21	การทำงานของ WEBSOCKET	32

สารบัญรูป (ต่อ)

รูปที่	หน้า
2.22 ตัวอย่างภาพรวมการใช้ DOCKER	34
2.23 องค์ประกอบพื้นฐานและการทำงานของ DOCKER	35
3.1 บล็อกไดอะแกรมการศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง	38
3.2 ภาพรวมการทำงานทั้งหมดของปริญญานิพนธ์	39
3.3 NETWORK TIMING DIAGRAM	40
3.4 บล็อกไดอะแกรมการออกแบบแบบจำลองสำหรับระบบถามตอบ	40
3.5 รูปแบบของชุดข้อมูล (DATASET FORMAT)	41
3.6 บล็อกไดอะแกรมของกระบวนการ PRE-PROCESSING	42
3.7 การดาวน์โหลดชุดข้อมูลสำหรับฝึก (TRAINING SET) ชุดข้อมูลประเมิน (VALIDATION SET) และชุดข้อมูลสำหรับทดสอบ (TEST SET) ด้วย LOAD_DATASET CLASS	43
3.8 ตัวแปร DATASETS ที่อยู่ในรูปแบบของ DATASETDICT	43
3.9 การสร้าง TOKENIZER OBJECT	44
3.10 การเพิ่มข้อมูลคำถามและบริบทใน TOKENIZED_DATASETS	44
3.11 การแบ่งข้อมูล (TRUNCATION) โดยกำหนด MAX_LENGTH = 416	45
3.12 การทับซ้อนกันของข้อความ (OVERLAPPING) โดยกำหนด STRIDE = 128	45
3.13 การทำข้อความให้ยาวเท่ากัน (PADDING) โดยกำหนด PADDING = "MAX_LENGTH"	46
3.14 การหา START_CHAR, END_CHAR, TOKEN_START_INDEX และ TOKEN_END_INDEX	47
3.15 การหา 'START_POSITION' และ 'END_POSITION'	48
3.16 OBJECT PARAMETERต่างๆที่ใช้ในการ OPTIMIZATION ด้วย OPTUNA	49

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.17	50
3.18	51
3.19	51
3.20	52
3.21	52
3.22	53
3.23	54
3.24	54
3.25	55
3.26	56
3.27	56
3.28	57
3.29	58
3.30	59
3.31	59
3.32	61
3.33	62
3.34	62

สารบัญรูป (ต่อ)

รูปที่	หน้า	
3.35	ท่าทางทักทายของตัวละคร	63
3.36	ท่าทางขยับเปิดปากของตัวละคร	63
3.37	ท่าทางขยับปิดปากของตัวละคร	64
3.38	ท่าทางของตัวละครที่มีลักษณะเปิดตา	64
3.39	ท่าทางของตัวละครที่มีลักษณะปิดตา	65
3.40	ตัวละครก่อนลบพื้นหลัง	65
3.41	ตัวละครหลังลบพื้นหลัง	66
3.42	หน้าเว็บไซต์ตอนยังไม่กดปุ่ม	67
3.43	หน้าเว็บไซต์เมื่อนำตัวชี้ตำแหน่ง (Cursor) ไปชี้ที่ปุ่ม	67
3.44	หน้าเว็บไซต์หลังจากกดปุ่ม	68
3.45	ส่วนประกอบต่าง ๆ ของหน้าต่าง CHAT-POPUP	68
3.46	หน้าต่าง CHAT-POPUP แสดงวิธีการใช้งานหลังกดปุ่ม “วิธีใช้งาน	69
3.47	หน้าต่าง CHAT-POPUP หลังกดปุ่ม “ASK ME QUESTION!” (ซ้าย) หน้าต่าง CHAT-POPUP หลังตัวการ์ตูนพูดทักทายจบ (ขวา)	71
3.48	หน้าต่าง CHAT-POPUP ในขณะที่มีการถามคำถามในช่องกรอกคำถาม	71
3.49	หน้าต่าง CHAT-POPUP ในขณะที่มีการแสดงคำถามและคำตอบ	72
3.50	ปุ่มบันทึกเสียง	73
3.51	หน้าต่าง CHAT-POPUP ในขณะที่บันทึกเสียง	73
3.52	หน้าต่าง CHAT-POPUP ในขณะที่มีการแสดงคำถามและคำตอบ	74
3.53	หน้าต่าง CHAT-POPUP กรณีที่ RECOMMENDATION SYSTEM API ไม่ สามารถใช้งานได้	75
3.54	บล็อกไดอะแกรมแสดงการทำงานของเว็บเซิร์ฟเวอร์	76
3.55	การสร้างสารบบใหม่สำหรับโปรเจค NODE.JS	76
3.56	การสร้างโปรเจคใหม่ในโพลเดอร์ NODE-WEB-SERVER	77
3.57	สร้างไฟล์ PACKAGE.JSON	77

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
3.58	ไฟล์ PACKAGE.JSON	78
3.59	การติดตั้งแพ็คเกจ EXPRESS และ SOCKET.IO	78
3.60	ข้อมูลในไฟล์ PACKAGE.JSON เมื่อติดตั้ง EXPRESS และ SOCKET.IO	79
3.61	ไฟล์ INDEX.JS สำหรับโปรแกรมสร้าง SERVER	79
3.62	โปรแกรมฝั่ง SERVER	80
3.63	เปิดพอร์ต (PORT) รอการเชื่อมต่อที่เบอร์ 5000	80
3.64	การตอบสนองของเซิร์ฟเวอร์เมื่อมี GET REQUEST ที่ URL HTTP://LOCALHOST:5000/	81
3.65	การเชื่อมต่อของฝั่งผู้ใช้บริการและผู้ให้บริการ	81
3.66	คำสั่ง SOCKET.ON() และ SOCKET.EMIT()	81
3.67	URL และข้อมูลของอ็อบเจกต์ OPTIONS ที่ใช้ในการ FETCH() API	82
3.68	คำสั่ง FETCH()	82
3.69	การติดตั้งแพ็คเกจ @GOOGLE-CLOUD/TEXT-TO-SPEECH	83
3.70	การเรียกใช้แพ็คเกจ @GOOGLE-CLOUD/TEXT-TO-SPEECH	83
3.71	การใช้งาน CLOUD TEXT-TO-SPEECH API	84
3.72	โปรแกรมฝั่งผู้ใช้งาน (CLIENT)	85
3.73	ผู้ใช้งานส่งคำขอ (REQUEST) มาที่เว็บเซิร์ฟเวอร์ HTTP://LOCALHOST:5000/	85
3.74	โปรแกรมฝั่งผู้ใช้งานในการส่งข้อมูลไปที่เว็บเซิร์ฟเวอร์	86
3.75	โปรแกรมฝั่งผู้ใช้งาน (CLIENT) ในการรับข้อมูลจากเว็บเซิร์ฟเวอร์	87
3.76	การสร้างแอปพลิเคชันใหม่บน HEROKU	88
3.77	การตั้งชื่อแอปพลิเคชัน	88
3.78	ขั้นตอนการ DEPLOY เว็บเซิร์ฟเวอร์	89
4.1	ข้อมูลตัวอย่างสำหรับการทดสอบกระบวนการเตรียมข้อมูล	93
4.2	เอาต์พุตก่อนทำ TRUNCATION	94

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่	หน้า
4.3	94
เอาต์พุตหลังทำ TRUNCATION กรอบสี่เหลี่ยม คือ พีเจอร์ที่ 1 และกรอบสี่เหลี่ยม คือ พีเจอร์ที่ 2	
4.4	95
เอาต์พุตหลังทำ OVERLAPPING กรอบสี่เหลี่ยม คือ พีเจอร์ที่ 1 และกรอบสี่เหลี่ยม คือ พีเจอร์ที่ 2	
4.5	95
เอาต์พุตหลังทำ PADDING กรอบสี่เหลี่ยม คือ พีเจอร์ที่ 1 และกรอบสี่เหลี่ยม คือ พีเจอร์ที่ 2	
4.6	96
เอาต์พุตหลังจากการทำ TOKENIZATION	
4.7	96
เอาต์พุตพีเจอร์ที่ 1 ของคู่คำถามและบริบท	
4.8	97
เอาต์พุตพีเจอร์ที่ 2 ของคู่คำถามและบริบท	
4.9	97
เอาต์พุตพีเจอร์ที่ 2 ของคู่คำถามและบริบท	
4.10	97
เอาต์พุตพีเจอร์ที่ 2 ของคู่คำถามและบริบท	
4.11	99
ค่าของ “START_POSITIONS” และ “END_POSITIONS” ทั้ง 2 FEATURE จากชุดข้อมูลตัวอย่างแสดงตำแหน่งของช่วงคำตอบ (ANSWER SPAN) ในบริบท (CONTEXT)	
4.12	99
คำตอบที่ได้จาก “START_POSITION” และ “END_POSITION” (LABELS GIVE ANSWER) เปรียบเทียบกับคำตอบจริงจากชุดข้อมูลตัวอย่าง (THEORETICAL ANSWER) ของพีเจอร์ที่ 1	
4.13	99
คำตอบที่ได้จาก “START_POSITION” และ “END_POSITION” (LABELS GIVE ANSWER) เปรียบเทียบกับคำตอบจริงจากชุดข้อมูลตัวอย่าง (THEORETICAL ANSWER) ของพีเจอร์ที่ 2	
4.14	100
HYPERPARAMETER จาก HYPERPARAMETER OPTIMIZATION	
4.15	100
ผลคะแนน EXACT MATCH และ F1	
4.16	101
ผลการทดสอบการเรียกดูข้อมูลบริบทตัวอย่าง จาก KNOWLEDGE BASE ตารางสำหรับเก็บข้อมูลบริบท	
4.17	101
ผลการทดสอบระบบด้วยคำถามที่ระบบไม่สามารถตอบได้ผ่าน POSTMAN	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.18	คำถามที่ระบบไม่สามารถตอบได้ ที่ถูกเพิ่มไปยัง KNOWLEDGE BASE ตารางสำหรับคำถามที่ระบบไม่สามารถตอบได้	102
4.19	ผลการทดสอบระบบตรวจสอบการหักทนาย ผ่าน POSTMAN	102
4.20	ผลการทดสอบระบบเตรียมข้อมูลคำถาม	103
4.21	ตัวอย่างบริบทก่อน-หลังผ่านระบบเตรียมข้อมูลบริบท	103
4.22	ผลการทดสอบระบบค้นคืนข้อมูล เพื่อหาบริบทที่เกี่ยวข้องกับคำถามมากที่สุด	104
4.23	ผลการทดสอบ READER MODEL	104
4.24	การทดสอบ API ของระบบโดยรวม ผ่าน POSTMAN โดยทดสอบด้วยข้อความหักทนาย	105
4.25	การทดสอบ API ของระบบโดยรวม ผ่าน POSTMAN โดยทดสอบด้วยคำถามนอกขอบเขตการตอบคำถามของระบบ	105
4.26	การทดสอบ API ของระบบโดยรวม ผ่าน POSTMAN โดยทดสอบด้วยคำถามที่เกี่ยวกับหลักสูตรภาควิชาวิศวกรรมคอมพิวเตอร์	106
4.27	การตอบสนอง (RESPONSE) จาก RECOMMENDATION SYSTEM API ที่ TERMINAL ของเว็บเซิร์ฟเวอร์	106
4.28	การตอบสนอง (RESPONSE) จาก GOOGLE CLOUD TEXT-TO-SPEECH API ที่ TERMINAL ของ WEB SERVER	107
4.29	หน้าเว็บไซต์ที่เว็บเซิร์ฟเวอร์ส่งกลับให้ผู้ใช้ (CLIENT)	107
4.30	TERMINAL ของเว็บเซิร์ฟเวอร์เมื่อได้รับข้อความจากผู้ใช้	108
4.31	หน้าเว็บไซต์หลังจากที่เว็บเซิร์ฟเวอร์ตอบกลับ	108
4.32	หน้าเว็บไซต์ขนาดความกว้างขั้นต่ำที่รองรับหรือขนาดความกว้าง 1366PX	109
4.33	หน้าเว็บไซต์ขนาดความกว้างที่ไม่รองรับหรือขนาดความกว้างน้อยกว่า 1366PX	110
4.34	หน้าเว็บไซต์ที่สามารถใช้งานได้ดีที่สุดขนาด 1920X1080 PX	110

สารบัญรูป (ต่อ)

รูปที่		หน้า
4.35	หน้าเว็บไซต์ที่ขนาด 1180X820PX (IPAD AIR)	111
4.36	หน้าเว็บไซต์ที่ขนาด 412X914PX (SAMSUNG GALAXY A51/71)	112
4.37	หน้าเว็บไซต์ของ NODEWEBSERVER.HEROKUAPP.COM	113
4.38	ผลลัพธ์เมื่อกดปุ่ม ASK ME QUESTION! และหน้า CONSOLE บนเว็บเบราว์เซอร์แสดงผลลัพธ์ข้อมูลไฟล์เสียง GREETING AUDIO	113
4.39	ผลลัพธ์เมื่อถามคำถามผ่านการพิมพ์ในช่องกรอกข้อความและหน้า CONSOLE บน WEB BROWSER แสดงผลลัพธ์ข้อมูลไฟล์เสียง ANSWER AUDIO	114
4.40	หน้าต่าง CHAT-POPUP ในขณะที่กำลังบันทึกเสียง	115
4.41	ผลลัพธ์เมื่อถามคำถามผ่านการใช้งาน SPEECH-TO-TEXT	115
4.42	ผลลัพธ์เมื่อทำการกดปุ่มวิธีใช้ (ซ้าย) และกลับสู่สถานะเดิมเมื่อกดปุ่มอีกครั้ง (ขวา)	116

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันหากต้องการทราบข้อมูลเกี่ยวกับภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังนั้นมีความยุ่งยากที่ต้องหาข้อมูลจากเล่มหลักสูตรที่มีจำนวนมาก ทำให้ผู้ใช้งานไม่มีความสะดวกและเสียเวลาในการค้นหา ปัญหานี้จึงเห็นปัญหาที่เกิดขึ้นจึงได้พัฒนาแบบจำลองที่ใช้สำหรับแนะนำข้อมูลเกี่ยวกับภาควิชาวิศวกรรมโทรคมนาคมได้เพื่อให้ทุกคนสามารถเข้าถึงได้ง่ายและใช้งานได้สะดวกมากยิ่งขึ้น ซึ่งการออกแบบการทำงานของแบบจำลองนี้ คือ ผู้ใช้งานป้อนคำถามที่เป็นเสียงหรือข้อความผ่านทางเว็บไซต์และระบบจะตอบกลับด้วย กราฟิกเคลื่อนไหวพร้อมเสียงและข้อความ โดยข้อมูลที่นำมาตอบอยู่ในฐานข้อมูล

1.2 วัตถุประสงค์

- 1) เพื่อศึกษาและพัฒนาระบบแนะนำข้อมูล (Recommendation System) ให้เป็นระบบที่มีการแนะนำเกี่ยวกับภาควิชาวิศวกรรมโทรคมนาคม สจล. โดยใช้ปัญญาประดิษฐ์ (Artificial Intelligence) และการประมวลผลภาษาธรรมชาติ (Natural Language Processing)
- 2) เพื่อศึกษาและพัฒนาการเขียนเว็บไซต์ (Website)
- 3) เพื่อศึกษาวิธีการสร้างกราฟิกเคลื่อนไหว
- 4) เพื่ออำนวยความสะดวกในการสอบถามเกี่ยวกับภาควิชาโทรคมนาคม สจล.

1.3 ขอบเขตของปัญญานี้

ศึกษาและพัฒนาระบบแนะนำข้อมูล (Recommendation System) ซึ่งแนะนำในเรื่องของภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง หลักสูตรปริญญาตรี โดยใช้ปัญญาประดิษฐ์ (Artificial Intelligence) และการประมวลผลภาษาธรรมชาติ (Natural Language Processing) ซึ่งจะทำการสร้างเป็นส่วนขยายเพื่อนำไปใช้บนเว็บไซต์ และสามารถโต้ตอบผ่านข้อความหรือเสียงพูด โดยแสดงผลเป็นกราฟิกเคลื่อนไหวออกมาด้วย

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

ปริญญาานิพนธ์เรื่อง “การศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง” ได้ทำการออกแบบและสร้างแบบจำลองของระบบแนะนำข้อมูล (Recommendation System) ซึ่งแบบจำลองนี้ใช้แนะนำและตอบคำถามในเรื่องของภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ดังนั้นปริญญาานิพนธ์ที่นำเสนอจึงมีหลักการที่เกี่ยวข้องดังต่อไปนี้

2.1 ระบบแนะนำข้อมูล (Recommendation System)

ระบบแนะนำข้อมูล (Recommendation System) [1] เป็นระบบที่แนะนำข้อมูล, สินค้าและบริการต่าง ๆ ให้กับผู้ใช้งาน โดยระบบจะพิจารณาแนะนำข้อมูล, สินค้าและบริการที่มีความคล้ายคลึงหรือมีส่วนที่เกี่ยวข้องกันมากที่สุดขึ้นมา ระบบแนะนำข้อมูล แบ่งออกเป็น 3 แบบหลัก ๆ ได้แก่

- 1) Content – based Filtering System แนะนำโดยพิจารณาที่เนื้อหาและคุณลักษณะของสินค้า หรือคำหลักต่าง ๆ
- 2) Collaborative Filtering System แนะนำโดยพิจารณาความคล้ายของพฤติกรรมผู้ใช้งาน

3) Hybrid System เป็นการผสมของทั้งสองแบบที่ได้กล่าวมาข้างต้น

ซึ่งการใช้งานระบบแนะนำข้อมูลนั้นจะต้องนำข้อมูลต่าง ๆ ไปผ่านกระบวนการเตรียมข้อมูลเบื้องต้น (Pre-processing) และการเทรนแบบจำลอง หลังจากผ่านการเทรนแบบจำลอง ข้อมูลแต่ละตัวจะมีค่าเวกเตอร์ของตัวเองที่ไม่เหมือนกัน กล่าวคือ คำที่มีความหมายใกล้เคียงกันจะมีค่าความคล้ายสูง ดังนั้น วิธีการแนะนำข้อมูล คือ หาความคล้ายกันของคำให้มากที่สุด โดยที่ผู้ใช้งานป้อนอินพุต คือ คำถาม เช่น ผู้ใช้งานถามคำถามว่า “ชื่อหลักสูตรคืออะไร” ระบบจะไปหาว่ามีข้อมูลใดบ้างที่มีความคล้ายกับ “ชื่อหลักสูตรคืออะไร” มากที่สุด ซึ่งการแนะนำข้อมูลจะมีความเหมาะสมมากน้อยนั้นก็ขึ้นอยู่กับความสัมพันธ์ของชุดข้อมูลที่ใช้ในการเทรนด้วย [1]

2.2 การประมวลผลภาษาธรรมชาติ (Natural Language Processing)

การประมวลผลภาษาธรรมชาติ (Natural Language Processing: NLP) เป็นวิทยาการแขนงหนึ่งที่ว่าด้วยเทคโนโลยีปัญญาประดิษฐ์ (Artificial Intelligence: AI) ซึ่งเป็นการทำให้คอมพิวเตอร์เข้าใจและสามารถที่จะสื่อสารด้วยภาษาปกติแบบที่มนุษย์ใช้สื่อสารได้ ซึ่งการประมวลผลภาษาธรรมชาตินำมาประยุกต์ใช้กับงานทางด้านการจำแนกข้อความ (Text Classification) หลากรูปแบบ เช่น ระบบรู้จำเสียงพูดทำการแปลงเสียงพูดเป็นข้อความ, การวิเคราะห์ความรู้สึกจากข้อความ และ แชทบอท ระบบพุดคุยถามตอบ [1]

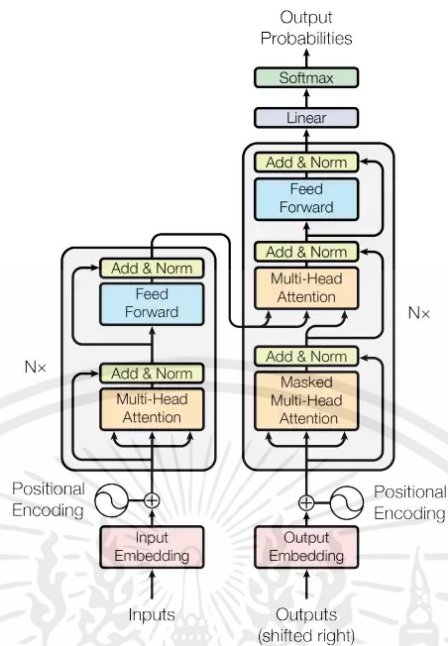
การประยุกต์ใช้งานของการประมวลผลภาษาธรรมชาตินั้นเป็นลักษณะการจำแนกข้อความ (Text Classification) แบบการเรียนรู้แบบมีผู้สอน (Supervised Machine Learning) ซึ่งการทำงานของงานการจำแนกข้อความนี้ประกอบไปด้วยขั้นตอน การนำข้อความไปหาค่าลักษณะ (Feature) จากนั้นก็นำไปเทรนเพื่อสร้างแบบจำลองขึ้นมา โดยกล่าวได้ว่าค่าลักษณะนี้ต้องเป็นในรูปแบบของค่าตัวเลขหรือค่าดัชนีบางอย่าง เพื่อให้คอมพิวเตอร์สามารถที่จะประมวลผลได้ ซึ่งการคำนวณค่าลักษณะนั้นก็มีหลายรูปแบบ เช่น TFIDF หรือ Word2Vec เป็นต้น [1]

2.3 Transformers Architecture

ในโลกของการเรียนรู้เชิงลึก (Deep Learning) ปัจจุบัน โครงสร้างเครือข่ายที่โดดเด่นและถูกใช้งานมากที่สุด คือ RNN (Recurrent Neural Network) และ CNN (Convolutional Neural Network) โดยทั้ง RNN และ CNN สามารถใช้ในงานด้านภาษาและวิสัยทัศน์ได้ ซึ่งทั้ง 2 วิธีนี้ยังมีประสิทธิภาพในการนำมาใช้งานได้ไม่ดีเท่าที่ควร เพื่อแก้ปัญหาในปี 2017 นักวิจัยจาก Google ได้เสนองานวิจัยโดยการยกเลิกใช้ทั้ง RNN และ CNN โดยได้เสนอสถาปัตยกรรมใหม่ที่มีชื่อว่า Transformers ซึ่ง Transformers คือ แบบจำลองทางบริบททางภาษารูปแบบหนึ่ง [2] ที่มีหัวใจหลักคือ กระบวนการที่เรียกว่า Self – Attention [3] โดยกระบวนการนี้นอกจากจะเป็นสิ่งที่ทดแทน RNN และ CNN ได้แล้วนั้นยังแสดงถึงความเกี่ยวข้องกันของคำต่าง ๆ ในข้อความด้วยตัวอย่างแบบจำลองในสถาปัตยกรรม Transformers เช่น GPT, BERT, BART และ T5 เป็นต้น

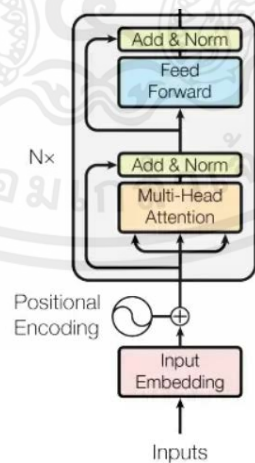
สถาปัตยกรรมทั่วไปของแบบจำลอง Transformers [4] ประกอบไปด้วย 2 ส่วนใหญ่ๆ ได้แก่ ส่วนการเข้ารหัส (Encoder) และ ส่วนถอดรหัส (Decoder) แสดงดังรูปที่ 2.1 [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 สถาปัตยกรรมทั่วไปของ Transformers [4]

จุดประสงค์ของส่วนการเข้ารหัส [5] คือ การดูลำดับของข้อมูลรับเข้า (Input) ทั้งหมดในคราวเดียวและตัดสินใจว่าโทเคน (Token) นี้มีความสอดคล้องกับโทเคนอื่น ๆ หรือไม่ โดยใช้ Attention Layer ซึ่งส่วนประกอบของส่วนการเข้ารหัสแสดงดังรูปที่ 2.2 [4]



รูปที่ 2.2 ส่วนประกอบของส่วนการเข้ารหัส [4]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.2 สามารถอธิบายส่วนประกอบต่าง ๆ เรียงจากล่างขึ้นบน ได้ดังนี้

1) ข้อมูลรับเข้า คือ ข้อมูลที่ถูก Tokenize แล้วซึ่งแสดงในรูปแบบของตัวเลขของลำดับสำหรับนำไปแปลงในขั้นต่อไปที่ต้องแสดงในรูปแบบของตัวเลขนั้น เนื่องมาจากว่าไม่สามารถใช้ข้อมูลในรูปแบบตัวอักษรในโครงข่ายประสาทเทียม (Neural Network) ได้ [5]

2) Input Embedding คือ ข้อมูลที่ถูก Tokenize แล้วมาแปลงเป็นรูปแบบของเวกเตอร์ (Vector) โดยที่แต่ละโทเคนมีขนาด 512 เช่น “I am a girl” ถูกแสดงในรูปแบบของ [1,2,3,4] หลังจากผ่าน Input Embedding Layers แล้วจะมีขนาด $4 * 512$ (1 เวกเตอร์ของแต่ละโทเคนมีขนาดเท่ากับ 512) ดังนั้นค่าแต่ละค่าที่มีความเกี่ยวข้องกันจึงมีค่าที่ใกล้เคียงกัน เช่น คำว่า “Police” กับ “Policeman” แต่คำว่า “Ploice” กับ “Water” จะไม่มีค่าที่ใกล้เคียงกัน [5]

3) Position Encoding คือ การระบุตำแหน่งของคำแต่ละคำ เนื่องจากว่าถ้าทำเพียงแค่ Input Embedding Layers จะได้แค่ค่าที่ใกล้เคียงกันเท่านั้นแต่ไม่ทราบตำแหน่งของคำแต่ละคำ เช่น “He has a black Ferrari. He also has a white cat.” กับ “He has a black & white Ferrari.” จากประโยค 2 ประโยคนี้สังเกตได้ว่าถ้าทำเพียงแค่ Input Embedding Layers คำว่า Black & White จะมีค่าความใกล้เคียงกันกับ Black และ White ของประโยคก่อนหน้า แต่ในความเป็นจริงแล้วคำทั้งสองคำนี้อยู่ไกลกันและมีบริบทที่แตกต่างกันโดยสิ้นเชิง จึงมีความจำเป็นที่จะต้องเพิ่ม Position Encoding เข้าไปเพื่อช่วยให้แบบจำลองสามารถรับรู้ตำแหน่งของคำที่กำลังพิจารณาอยู่ โดย Transformers เลือกใช้ค่า sin และ cos ในการแทนตำแหน่งต่างๆ ดังสมการที่ 2.1 และ 2.2 [5]

$$PE_{(pos, 2i)} = \sin \left(\frac{pos}{\frac{2i}{10000^{\frac{d_{model}}{2}}}} \right) \quad (2.1)$$

$$PE_{(pos, 2i+1)} = \cos \left(\frac{pos}{\frac{2i}{10000^{\frac{d_{model}}{2}}}} \right) \quad (2.2)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยที่	pos	คือ ตำแหน่ง
	d_{model}	คือ ขนาดของ Embedding ในแต่ละโทเคน (Token)
	$2i, 2i+1$	คือ แต่ละขนาดของ Position Vectors โดย $2i$ คือ $0, 2, 4, \dots, 512$ และ $2i+1$ คือ $1, 3, 5, \dots, 511$

4) แกนของส่วนเข้ารหัส ที่ประกอบด้วย 2 ส่วนหลักๆ คือ Head Attention แบบ Multi – Head Attention ซึ่งเป็น Sub – Layer ซึ่งแต่ละลูกศรคือ V, K, Q หลัก และ Fully Connected Neural Network แบบ Position-Wise Fully Connected Feed-Forward Network เรียกเป็น Position-wise เพราะ Transformers จะใช้ Networks อันเดียวกันกับทุกๆ คำในประโยค คล้ายกับ RNN ก็คือในตอนคำนวณแต่ละคำจะถูกคูณด้วย Weight Matrix เดียวกันทั้งหมด ซึ่งจะลดจำนวนพารามิเตอร์ ที่แบบจำลองต้องเรียนรู้ไปได้มาก พร้อม ๆ กันกับควบคุมไม่ให้แบบจำลองยึดติดไปกับตำแหน่งใดตำแหน่งหนึ่งของคำแค่ตำแหน่งเดียว [5]

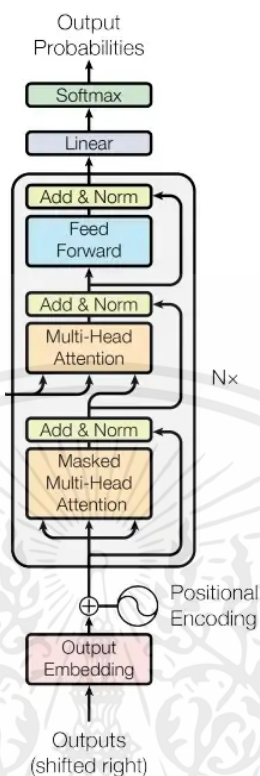
จุดประสงค์ของส่วนถอดรหัส คือ การกำหนดโทเคนของลำดับเอาต์พุต (Output) ที่ละรายการ ซึ่งส่วนประกอบของส่วนการถอดรหัสแสดงดังรูปที่ 2.3 [4]

จากรูปที่ 2.3 สามารถอธิบายส่วนประกอบต่าง ๆ เรียงจากล่างขึ้นบน ได้ดังนี้

1) เอาต์พุต คือ การแสดงตัวเลขของลำดับที่ถูก Tokenize เช่นเดียวกับที่ทำในส่วนการเข้ารหัสแต่มีความแตกต่างกันที่เอาต์พุตมีการแสดงตัวเลขที่เลื่อนไปทางขวา เนื่องจากส่วนถอดรหัสถูกเทรนให้ทำนายคำต่อไปของลำดับจากโทเคนก่อนหน้า ซึ่งโทเคนลำดับแรกมักจะมีปัญหาเพราะไม่มีโทเคนก่อนหน้า ดังนั้นแก้ปัญหาด้วยเลื่อนลำดับของเอาต์พุตและมีการแทรก BOS (Beginning of Sentence) ไว้ที่จุดเริ่มต้นของประโยค [5]

2) Output Embedding & Positional Encoding Layers มีหน้าที่และโครงสร้างเช่นเดียวกับส่วนเข้ารหัส [5]

3) แกนของส่วนถอดรหัส ประกอบด้วยบล็อกเรียงกัน 6 ชั้นโดยมีการเพิ่ม Masked Multi – Head Attention เข้ามา โดยผลลัพธ์จากส่วนเข้ารหัสบล็อกสุดท้ายถูกส่งเข้ามาที่ Layer ตรงกลาง และผลลัพธ์ที่ออกจากบล็อกสุดท้ายของส่วนถอดรหัสจะถูกนำเข้าไป Softmax เพื่อเป็นเอาต์พุตต่อไป [4]



รูปที่ 2.3 ส่วนประกอบของส่วนการถอดรหัส [4]

2.3.1 Attention Module

การนำคำทุกคำในประโยคมาเปรียบเทียบกับกันเอง ให้แบบจำลองเรียนรู้และเลือกเองว่าจะสนใจคำไหน เมื่อไหร่ ด้วยการแปลงอินพุตเป็น 3 เวกเตอร์ คือ Q (Query), K (Key) และ V (Value) โดยการคำนวณ Q, K และ V จะคำนวณจากการคูณอินพุตเข้ากับค่าน้ำหนัก (Weight) จะได้เป็น WQ , WK , WC หรือคือ Linear Layer โดยค่าน้ำหนักเหล่านั้นจะกลายเป็น Learned Parameter ที่แบบจำลองจะเรียนรู้ขึ้นมาเอง จากการเทรนด้วย Gradient Descent ตามปกติ กระบวนการนี้เรียกว่า Self-Attention [6] การทำงานจะคล้ายกับการทำงานของสมองเพื่อการแยกแยะสิ่งที่กำลังสนใจ ดังนั้นสิ่งที่เกิดขึ้นคือ ไม่จำเป็นต้องดูทุกองค์ประกอบเพื่อตัดสินใจ เพียงแค่เลือกแค่ข้อมูลบางจุดที่สำคัญเท่านั้น [7] โดย Attention ใน Transformers จะต่างจาก Attention ใน Sequence-to-Sequence หลายประการ ประการแรกคือไม่ใช่เป็นการหา Attention Vector ของส่วนถอดรหัส จากส่วนเข้ารหัสเพียงทางเดียวเท่านั้น แต่ทุก ๆ หน่วยของข้อมูลจะทำ

Attention กันเองด้วย สิ่งนี้เรียกว่า Self-Attention ดังที่ได้กล่าวมาข้างต้น และการทำ Attention จะเป็นแบบ Scaled Dot-Product Attention ที่เสริมด้วย Multi-Head Attention [8]

Scaled Dot-Product Attention โดยเริ่มจากการแบ่งประเภทเวกเตอร์เป็น Query (Q), Key (K) และ Value (V) โดย K คือ เวกเตอร์ที่นำไปทำ Dot Product กับ Q เพื่อดูว่าควรจะใส่ใจกับตำแหน่งไหน และ V จะเป็นค่าที่นำมาใช้ เปรียบเสมือนมีกล่อง (k, v) อยู่หลายใบ โดย k เป็นแม่กุญแจล็อกคอก และ v เป็นของที่อยู่ข้างใน ส่วน q ก็คือลูกกุญแจที่นำมาไขกล่อง ถ้า q เข้ากับ k ได้ดี จะสามารถนำ v ไปใช้ได้มาก ซึ่งการทำ Attention นี้คือฟังก์ชัน A ที่รับ q กับ K และ V เข้ามาแล้วให้ผลลัพธ์ออกไปเป็นเวกเตอร์ตัวหนึ่งที่เป็นส่วนผสมของ v ต่างๆ โดยให้ความสำคัญกับ v ตามค่าของ k·q ดังสมการที่ 2.3

$$A(q, K, V) = \sum_i \frac{e^{k_i \cdot q}}{\sum_j e^{k_j \cdot q}} v_i \quad (2.3)$$

หากต้องการคำนวณ q หลาย ๆ ตัว สามารถทำให้เป็นการคูณเมทริกซ์ ดังสมการที่ 2.4

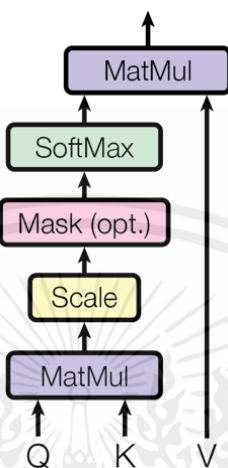
$$A(Q, K, V) = \text{softmax}(QK^T)V \quad (2.4)$$

โดยที่แต่ละเวกเตอร์ q, k, v คือ Row Vector ใน Q, K, V ตามลำดับ และการทำ SoftMax ในที่นี้จะทำตามแถวและผล Attention Vector ของแต่ละ q ซึ่งก็คือแต่ละแถวใน A(Q, K, V) ทั้งนี้หากขนาดของ k ซึ่งเท่ากับขนาดของ q มีค่าสูง ค่าของ k·q ก็สูงตามไปด้วย ซึ่งพอเข้า SoftMax แล้วจะไปอยู่ในบริเวณที่มี Gradient น้อย ทำให้ยากต่อการหาค่าที่เหมาะสมที่สุด จึงได้มีการลดขนาดของ k·q ตามจำนวนมิติของ k ด้วย ดังสมการที่ 2.5

$$A(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (2.5)$$

การใช้ Attention แบบ Dot Product ที่สามารถประหยัดเวลาและเนื้อที่ในการคำนวณมากกว่า Attention แบบอื่น ๆ ทำได้โดยการเพิ่ม Scaling Factor เข้าไป จะได้ออกมาเป็น

Scaled Dot-Product Attention ซึ่งอธิบายได้ด้วยสมการที่ 2.5 ข้างต้น สามารถเขียนเป็นแผนภาพได้ดังรูปที่ 2.4 [8]



รูปที่ 2.4 แผนภาพ Scaled Dot-Product Attention [8]

Multi-Head Attention โดยจาก Scaled Dot-Product Attention ถ้านำมาใช้เลยทันทีจะมีข้อจำกัดอยู่สองอย่าง ข้อแรก คือ เป็นการนำค่า Q, K, V มาคำนวณตรงๆ ซึ่งอาจจะไม่ใช่วิธีการที่เหมาะสมสำหรับการทำ Attention ควรจะให้ระบบมีการเรียนรู้วิธีการที่เหมาะสมด้วยการใส่ค่าน้ำหนักเข้าไป ทำให้ Attention Module อยู่ในรูปแบบดังนี้

$$A(QW^Q, KW^K, VW^V)$$

อย่างไรก็ตามก็ยังมีข้อจำกัดอยู่ตรงที่ทุกหน่วยของข้อมูลมีปฏิสัมพันธ์กันได้เพียงแบบเดียว ซึ่งไม่เพียงพอสำหรับงานทั่วไปที่หน่วยต่าง ๆ ของข้อมูลจะมีความสัมพันธ์กันได้หลายแบบ เช่น ประโยคที่ว่า “ฉันเลี้ยงแมว” คำว่า “แมว” กับ “ฉัน” อาจจะมีความสัมพันธ์โดยที่ “แมว” เป็นกรรมตรงของประโยคที่มีประธานคือ “ฉัน” หรือ “ฉัน” คือสิ่งที่แสดงความเป็นเจ้าของของ “แมว” ก็ได้ เพราะฉะนั้น จึงควรมี Q, K, V ไว้หลายๆ ชุด ซึ่งจะถูคูณด้วยค่าน้ำหนักที่ต่างกันออกไป เพื่อให้สามารถเรียนรู้ความสัมพันธ์ต่าง ๆ ได้ จากนั้นจึงทำ Attention ในแต่ละชุดก่อนที่จะรวมทุกชุดเข้าด้วยกันเป็นข้อมูลเพื่อส่งไปประมวลผลต่อไป โดย Attention ชุดหนึ่ง จะเรียกว่าหัวหัวหนึ่ง

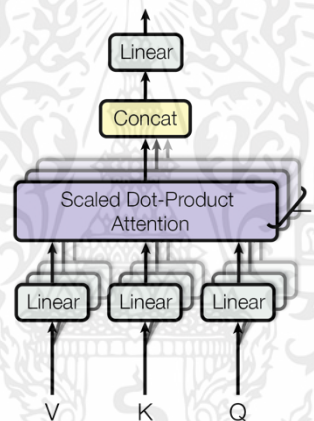
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังสมการที่ 2.6 และ Multi-Head Attention คือการนำทุกหัวมารวมกัน ซึ่งเขียนเป็นสมการได้ดังสมการที่ 2.7

$$\text{head}_i = A(QW_i^Q, KW_i^K, VW_i^V) \quad (2.6)$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \text{head}_2, \dots, \text{head}_h)W^O \quad (2.7)$$

การที่มีหลายหัวเช่นนี้ อาจเทียบได้กับ CNN ที่มีหลายช่องซึ่งแต่ละช่องก็จะเรียนค่าลักษณะที่แตกต่างกันไป สุดท้ายแล้ว Multi-Head Attention สามารถสรุปออกมาเป็นแผนภาพแสดงดังรูปที่ 2.5 [8]



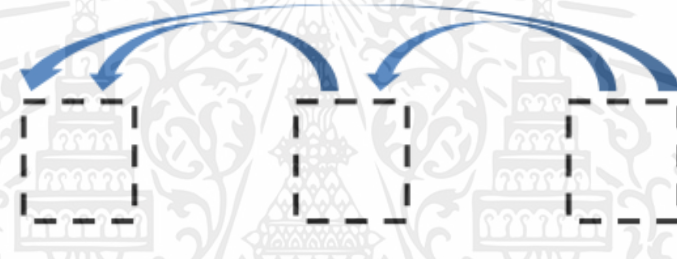
รูปที่ 2.5 แผนภาพ Multi-Head Attention [8]

ใน Transformers จะไม่ได้ใช้แค่เพียง Self-Attention อย่างเดียว แต่จะใช้ Attention ที่ต่างกันสามแบบ แบบแรกคือ Self-Attention ปกติตั้งที่กล่าวมาข้างต้นที่ทุกหน่วยจะทำ Attention กับทุกหน่วย แสดงดังรูปที่ 2.6 [8] แบบที่สองคือ Masked Self-Attention ที่แต่ละหน่วยทำ Attention กับเฉพาะข้อมูลที่อยู่ด้านหน้าเท่านั้น โดย Attention ชนิดนี้จะใช้กับส่วนถอดรหัส เนื่องจากกระบวนการสร้างเอด์พูดในตอนใช้งานจริงเป็นการสร้างทีละตัว จึงไม่สามารถนำข้อมูลจากอนาคตมาใช้ได้ แสดงดังรูปที่ 2.7 [8] แบบที่สามคือ Attention ที่เหมือนกับที่ใช้ใน

Sequence-to-Sequence ทั่วไป โดยฝั่งถอดรหัสจะเป็นตัว Query ข้อมูลจากฝั่งเข้ารหัส แสดงด้วยรูปที่ 2.8 [8]



รูปที่ 2.6 Self-Attention [8]



รูปที่ 2.7 Masked Self-Attention [8]



รูปที่ 2.8 Encoder-Decoder Attention [8]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 Bidirectional Encoder Representations from Transformers: BERT

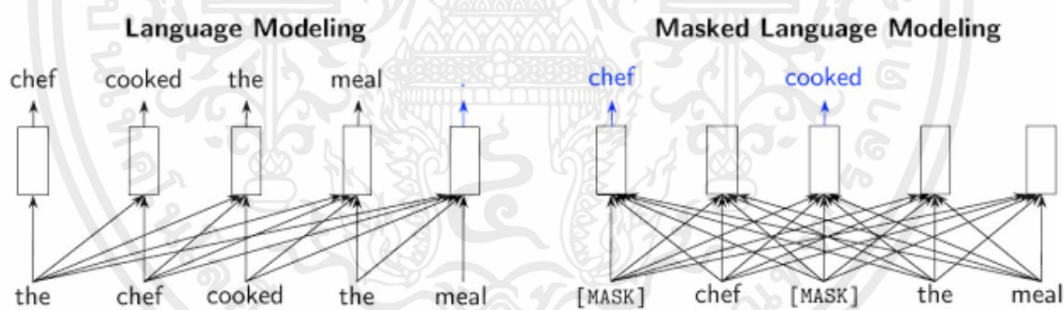
BERT เป็นแบบจำลองทางภาษา (Language Model) หนึ่งในที่มีโครงสร้างมาจากส่วนเข้ารหัส ของ Transformers ซึ่งจะแตกต่างกันตรงที่ BERT จะขยายให้มีจำนวน Attention Head มากขึ้น มีจำนวน Layers มากขึ้น และเพิ่มขนาดของ Embedding Vectors

2.4.1 Pretraining BERT

ทางทีม Google AI ได้เปลี่ยนโจทย์แบบจำลองทางภาษาแบบดั้งเดิม คือเรียนรู้จากบริบท (Context) ที่มาจากคำที่อยู่ทางซ้ายและขวา มาเป็นการเรียนรู้โดยไม่มีผู้สอน (Unsupervised Learning) ด้วยกัน 2 โจทย์ คือ Masked Language Model: MLM และ Next Sentence Prediction: NSP

2.4.1.1 Masked Language Model: MLM

MLM ถูกออกแบบมาเพื่อใช้แทนที่แบบจำลองทางภาษาแบบดั้งเดิม โดยในแต่ละรอบของการสอนแบบจำลอง ดังแสดงในรูปที่ 2.9 [9]



รูปที่ 2.9 Language Model แบบดั้งเดิม (ซ้าย) เปรียบเทียบกับ Masked Language Model (ขวา) [9]

เมื่อประโยคจำนวนหนึ่งจะถูกป้อนเข้ามาในแบบจำลอง จะเลือกมา 15% ของคำทั้งหมด จากคำทั้งหมดที่เลือกมานั้น 80% จะถูกแทนที่ด้วย [MASK] 10% จะถูกแทนที่ด้วยคำอื่น ๆ แบบสุ่ม และอีก 10% จะไม่มีการเปลี่ยนแปลง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

80% ที่ถูกแทนที่ด้วย [MASK] นั้น แบบจำลองจะพยายามทำนาย (Predict) คำตอบเติม โดยพิจารณาจากคำรอบ ๆ ในบริบท

2.4.1.2 Next Sentence Prediction: NSP

ในกระบวนการเทรน BERT แบบจำลองจะได้รับคู่ของประโยคเป็นอินพุต และเรียนรู้ที่จะทำนายว่าประโยคที่สองในคู่นั้นเป็นประโยคที่อยู่ติดกันในเอกสารต้นฉบับ ในระหว่างการเทรน 50% ของอินพุตเป็นคู่ประโยคที่อยู่ติดกันในข้อมูลต้นฉบับ อีก 50% จะเลือกประโยคสุ่มจากคลังข้อมูลเป็นประโยคที่สอง

2.4.2 Fine-tuning BERT

Fine-tuning นั้นมีความตรงไปตรงมา เนื่องจาก Self-Attention Mechanism ของ Transformers ช่วยให้ BERT สามารถสร้างแบบจำลองกับงานต่าง ๆ ในด้านการประมวลผลภาษาธรรมชาติ มากมาย สำหรับการประยุกต์ใช้ที่เกี่ยวข้องกับคู่ข้อความ โดยทั่วไปจะมีรูปแบบคือ เข้ารหัสคู่ข้อความนี้ก่อนจะใช้ Bidirectional Cross Attention

BERT ใช้ Self-Attention Mechanism เพื่อรวมข้อความที่สนใจกับ Self-Attention อย่างมีประสิทธิภาพ โดยรวม Bidirectional Cross Attention ไประหว่างทั้ง 2 ประโยค

สำหรับแต่ละงานจะต้องทำอินพุตและเอาต์พุตเฉพาะงานเข้ากับ BERT และปรับแต่งพารามิเตอร์ตั้งแต่ต้นจนจบ อย่างงานของระบบถามตอบ (Question Answering) อินพุตจะเป็นรูปแบบของคู่คำถามและบริบท (Question And Context Paired) [10]

2.5 Open-domain Question Answering: ODQA

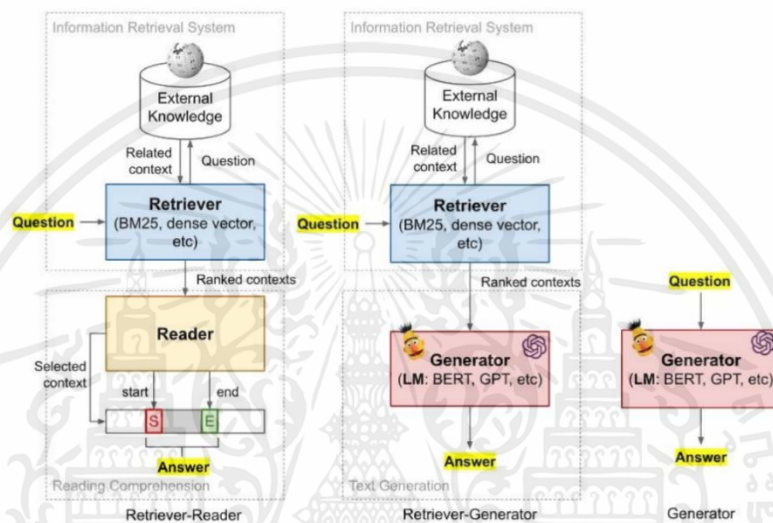
ODQA เป็นงานของภาษาประเภทหนึ่งซึ่งจะให้มีการตั้งคำถามที่เป็นภาษาธรรมชาติ (Natural Language) ให้กับแบบจำลองเพื่อให้แบบจำลองสร้างคำตอบที่ถูกต้องออกมา โดย ODQA นั้นสามารถทำงานได้โดยมีหรือไม่มีแหล่งข้อมูลจากภายนอกก็ได้ ซึ่งเงื่อนไขทั้งสองแบบนี้ เรียกว่า Open – book หรือ Close – book Question Answering

ODQA สามารถแบ่งประเภทได้เป็น 3 รูปแบบ ดังนี้และแสดงดังรูปที่ 2.10 [11]

1) Open – book QA: Retriever – Reader แบบจำลองสามารถจดจำ และตอบสนองได้อย่างถูกต้องด้วยคำตอบของคำถามที่ให้เห็นในขณะฝึกสอน (Training)

2) Open – book QA: Retriever – Generator แบบจำลองสามารถ ตอบคำถามใหม่ได้ในตอนที่ทดสอบ (Test) และเลือกคำตอบจากชุดคำตอบที่เห็นระหว่างการฝึกสอน (Training)

3) Close – book QA: Generative Language Model แบบจำลองสามารถตอบคำถามใหม่ ๆ ที่มีคำตอบที่ไม่มีอยู่ในชุดข้อมูลการฝึกสอน (Training)



รูปที่ 2.10 ภาพรวมการทำงานของ Open-domain Question Answering (ODQA) ทั้ง 3 แบบ [11]

ในปริณญาณิพนธ์นี้ ผู้จัดทำได้เลือกใช้ Open-book QA: Retriever – Reader ในการสร้างระบบ

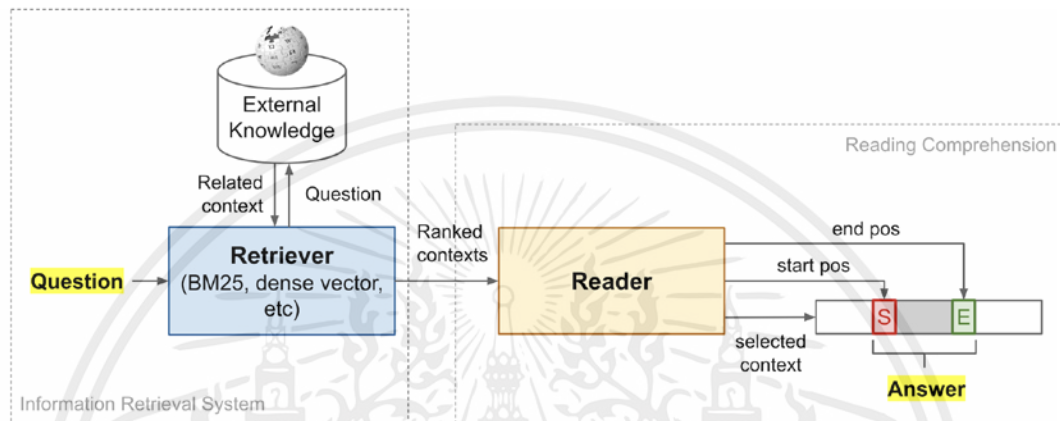
2.5.1 Open-book QA: Retriever-Reader

ถ้าแบบจำลองทางภาษาไม่มีบริบท หรือไม่มีหน่วยความจำเพียงพอในการเก็บบริบทจากชุดข้อมูลสำหรับฝึก (Training Set) จะไม่สามารถทำนายคำตอบที่ถูกต้องได้ เปรียบเสมือนกับการสอบแบบเปิดตำราได้ ที่นักศึกษาจะได้รับอนุญาตให้สืบค้นจากแหล่งความรู้ภายนอก อย่างหนังสือหรือสมุดบันทึกบทเรียนในขณะที่ทำข้อสอบ ระบบ Open-domain Question Answering ก็เช่นเดียวกัน ระบบนี้จะมาคู่กับฐานความรู้ (Knowledge Base) เพื่อค้นหาเอกสารที่เกี่ยวข้อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการสกัดคำตอบจากบริบท สามารถแบ่งกระบวนการในการหาคำตอบ ออกเป็น 2 ขั้นตอน ดังแสดงในรูป 2.11 [11] คือ

- 1) หาบริบทที่เกี่ยวข้อง (Related Context) จากฐานความรู้ด้วย Retriever Model
- 2) ทำกระบวนการสกัดเอาคำตอบจากบริบทที่เกี่ยวข้องด้วย Reader



รูปที่ 2.11 โครงสร้างของระบบ Open-book QA: Retriever-Reader [11]

2.6 Retrievers for Question-Answering System

Retriever จะถูกใช้ในการดึงค่าลักษณะจากคำถามไปค้นหาบริบทที่เกี่ยวข้องจากฐานความรู้ แล้วส่งคืนบริบทที่เกี่ยวข้องนี้กลับมาเพื่อนำไปสกัดเอาคำตอบด้วย Reader Model ในกระบวนการถัดไป

BM25 เป็นเครื่องมือค้นหา (Search Engine) ชนิดหนึ่งที่มีการพัฒนามาจากแนวคิดของ Term Frequency-inverse Document Frequency: TF-IDF เพื่อปรับปรุงและพัฒนาในส่วนของความอิ่มตัวของคำ (Term Saturation) และความยาวของเอกสาร (Document Length) [12]

ความอิ่มตัวของคำ (Term Saturation) สามารถอธิบายได้ว่า หากต้องการค้นหาคำว่า “หลักสูตรโทรคม” นั่นคือมีคำว่า “หลักสูตร” และ “โทรคม” แสดงว่าต้องการคำว่า “หลักสูตร” และ “โทรคม” อย่างละ 1 คำ ไม่ได้ต้องการคำว่า “หลักสูตร” หรือ “โทรคม” เพียงอย่างเดียวหนึ่ง หากใช้วิธีของ Term Frequency: TF แบบปกติคือการนับคำแต่ละคำที่ต้องการในบริบทที่เกี่ยวข้อง จากคำว่า “หลักสูตรโทรคม” จะได้ “หลักสูตร” และ “โทรคม” มีค่า TF เท่ากับ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีของ BM25 จะสมมติให้ค่าพารามิเตอร์ $k = 1$ ดังนั้นทั้งคำว่า “หลักสูตร” และ “ไทรคอม” แต่ละคำจะมีค่า $TF/(TF+1) = 1/2$ แต่หากเจอคำว่า “หลักสูตรหลักสูตร” หรือ “ไทรคอมไทรคอม” แต่ละคำจะมีค่า $TF = 2$ และจะได้ค่า $TF/(TF+1) = 2/3$ จะเห็นว่าค่าที่เกิดขึ้นเพียงครั้งเดียวมีค่าของความน่าจะเป็นไปได้มากกว่าค่าเดียวกันแล้วเกิดขึ้นสองครั้ง

จะให้ค่าเอกสารสั้น ที่มีคำที่ต้องการหามากกว่าเอกสารยาวแต่ไม่มีคำที่ต้องการ หากต้องการทราบว่าเอกสารนั้นสั้นหรือยาวก็ให้หาค่าเฉลี่ยของความยาวของเอกสารแต่ละอันมาเป็นความยาวอ้างอิง โดยในเงื่อนไขนี้จะมีพารามิเตอร์ b เป็นพารามิเตอร์หลัก กล่าวคือ หากความยาวของเอกสาร (dl) มีความยาวยาวกว่าความยาวค่าเฉลี่ย เอกสารนี้จะได้รับการจัดอันดับคะแนนที่ต่ำกว่าและพารามิเตอร์ b ก็จะถูกปรับให้มีค่าสูงขึ้น แต่ถ้าความยาวน้อยกว่าความยาวเฉลี่ยค่าพารามิเตอร์ b ก็จะถูกปรับให้มีค่าลดลง ซึ่งค่าพารามิเตอร์ b จะมีค่าอยู่ระหว่าง $0 < b < 1$

จากที่กล่าวมา BM25 เป็นการพัฒนามาจาก TF - IDF เพื่อการปรับปรุงที่ดียิ่งขึ้นในการให้ความสำคัญของความยาวเอกสารและความอิมิตัวของความถี่ของคำนั้น ๆ [13] ดังสมการที่ 2.8

$$W_{i,j} = \left(\frac{f_{i,j}(k_1+1)}{f_{i,j} + k_1(1-b) + b \left(\frac{dl}{dl_{avg}} \right)} \right) \times idf \quad (2.8)$$

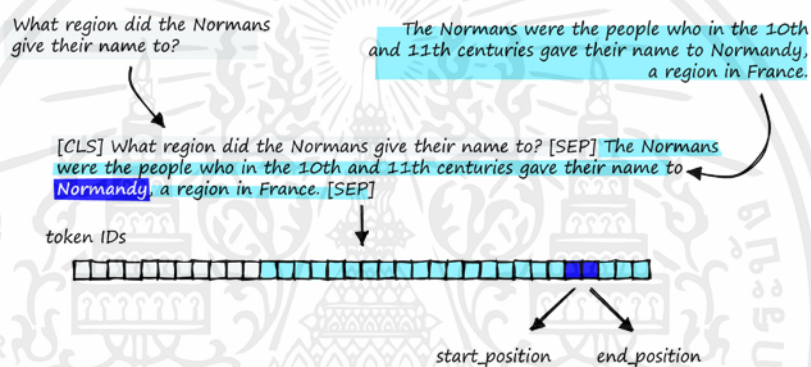
โดยที่	$f_{i,j}$	คือ ค่าของ TF
	b	คือ ค่าปรับ Normalization ค่าอยู่ระหว่าง $0 < b < 1$
	k_1	คือ ค่าที่ใช้ในการปรับพารามิเตอร์ โดยที่ $k_1 > 0$
	dl	คือ ค่าความยาวของเอกสารนั้นๆ
	dl_{avg}	คือ ค่าเฉลี่ยของความยาวของเอกสารนั้นๆ
	idf	คือ ค่าของ IDF

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 Reader Model for Question-Answering System

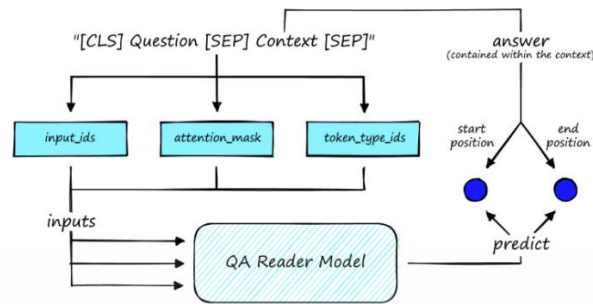
Reader Model เป็นส่วนสุดท้ายของ ODQA ถูกสร้างมาจากการทำ Fine-tuning Pre-trained Model จะใช้ในการอ่านบริบทที่ถูกส่งคืนมาจาก Retriever Model และจะทำการสกัดคำตอบจากบริบทนี้

สำหรับการสกัดคำตอบ จะไม่ใช้การหาคำตอบที่เป็นข้อความออกมาตรง ๆ แต่เป็นการที่แบบจำลองจะอ่านค่าโทเคนของ “input_ids” [14] และจะคืนค่าช่วงคำตอบ (Answer Span) จากตำแหน่งเริ่มต้นหรือโทเคน “start_position” และตำแหน่งสิ้นสุดของคำตอบหรือโทเคน “end_position” ของ “input_ids” ดังแสดงในรูปที่ 2.12 [14]



รูปที่ 2.12 “input_ids” ที่ถูกถอดรหัส โดยสีเทาแสดงคำถาม สีฟ้าแสดงบริบท และสีน้ำเงินแสดงคำตอบ [14]

สำหรับการเทรน Reader Model โดยใช้ BERT Pre-trained Model จะต้องการ 2 อินพุตและ 2 Labels คือ คู่คำถามและบริบท (Question And Context Paired) และตำแหน่งเริ่มต้นและสิ้นสุดของคำตอบ ตามลำดับ เริ่มต้นจะนำคู่คำถามและบริบทไปทำกระบวนการเตรียมข้อมูล (Pre-processing) เพื่อแปลงเป็น “input_ids”, “attention_mask” และโทเคน “start_position” และ “end_position” เพื่อป้อนเข้าสู่ Reader Model ต่อไป แสดงโครงสร้างการทำงานดังในรูปที่ 2.13

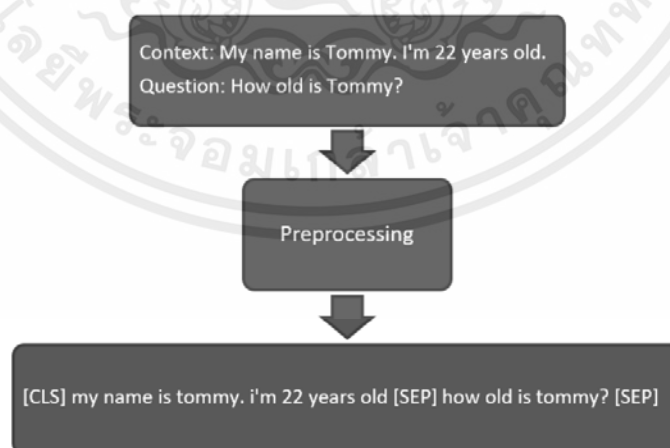


รูปที่ 2.13 โครงสร้างการทำงานของ Reader Model [14]

2.7.1 กระบวนการเตรียมข้อมูล (Pre-processing)

จุดมุ่งหมายของกระบวนการนี้คือเพื่อแปลงคู่คำถามและบริบท (Question And Context Paired) เป็นข้อมูลที่จะสามารถป้อนเข้าสู่ Pre-trained Model เพื่อที่จะทำการเทรนได้ ดังแสดงในรูปที่ 2.14 ซึ่งจะมีขั้นตอนที่สำคัญคือ Tokenization และการหาตำแหน่งของช่วงคำตอบ (Answer Span)

เอาต์พุตสุดท้ายที่ต้องการจากกระบวนการเตรียมข้อมูลนี้ จะอยู่ในรูปแบบ Dictionary ที่ประกอบไปด้วย Key “input_ids”, “attention_mask”, “start_positions” และ “end_positions” ดังแสดงในรูปที่ 2.15



รูปที่ 2.14 คู่คำถามและบริบทก่อนและหลังผ่านกระบวนการเตรียมข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
dict_keys(['input_ids', 'attention_mask', 'start_positions', 'end_positions'])
```

รูปที่ 2.15 Dictionary Key ของเอาต์พุตหลังผ่านกระบวนการเตรียมข้อมูล

2.7.1.1 Tokenization

Tokenizer เป็นเครื่องมือที่สำคัญและเป็นหัวใจหลักของกระบวนการเตรียมข้อมูล โดย Tokenizer จะทำหน้าที่แยกคำออกจากข้อความ (Tokenize) ให้เป็นตัวเลขหรือโทเคน (Token) ตามกฎเกณฑ์ของ Tokenizer ที่เลือกใช้ [15] นอกจากนี้จะแยกคำออกจากข้อความแล้ว Tokenizer ยังทำหน้าที่เติมโทเคนพิเศษ (Special Token) ให้กับคู่คำถามและบริบทด้วย สำหรับ Tokenizer ที่ใช้ในกระบวนการเตรียมข้อมูล สามารถเรียกใช้ได้จาก Pre-trained Model ที่จะนำมา Fine-tuning หรือจะสร้าง Tokenizer ขึ้นมาเองก็ได้ ซึ่งจะมีด้วยกัน 3 รูปแบบ ดังนี้

1) Word-based Tokenization

การ Tokenize แบบนี้เป็นวิธีที่ง่ายที่สุดและตรงตัวมากที่สุด เพราะจะแปลงคำที่เจอเป็นโทเคนโดยตรง แต่ปัญหาที่ตามมาคือวิธีการนี้มีความเสี่ยงสูงที่บางคำจะหายไปจากชุดข้อมูลสำหรับฝึก เช่น ถ้าแบบจำลองเคยเห็นคำว่า “foot” และ “ball” จากชุดข้อมูลสำหรับฝึก แต่เมื่อไปเจอคำว่า “football” แบบจำลองจะไม่สามารถตรวจจับได้ [16]

2) Character-based Tokenization

เป็นการ Tokenize โดยที่แยกข้อความออกมาเป็นตัวอักษร (Character) ซึ่งจะเป็นตัวอักษรที่สามารถถอดรหัสได้ด้วย ASCII หรือ Unicode ข้อเสียที่เห็นได้ชัดเลยคือ ถ้าให้ข้อความยาวกับ Character-based Tokenization ต้องใช้หน่วยประมวลผลที่มากเมื่อเทียบกับ Word-based Tokenization และจะสูญเสียความหมายไปด้วย

3) Subword Tokenization

Subword Tokenization คือรูปแบบที่สร้างขึ้นมาเพื่อแก้ไข ปัญหาของ Word-based และ Character-based Tokenization เป็นรูปแบบที่จะ Tokenize ตามกฎที่ได้สร้างขึ้น Subword Tokenization จะ Tokenize ให้เป็นคำย่อยที่มีความหมาย เช่น คำว่า “boy” จะไม่ถูก Tokenize แต่คำว่า “boys” จะถูกแยกเป็น “boy” และ “s” ซึ่งจะช่วยให้แบบจำลองเข้าใจคำว่า “boys” เกิดมาจากคำว่า “boy” ที่มีความหมายเปลี่ยนไป แต่เกิดจากราก

ศัพท์เดียวกัน ซึ่ง Subword Tokenization จะมีอัลกอริทึม (Algorithm) ด้วยกัน 3 ชนิด คือ Byte-Pair Encoding (BPE), WordPiece, และ SentencePiece

เมื่อนำคู่คำถามและบริบทไปทำ Tokenization เอาต์พุตที่ได้ออกมาจะอยู่ในรูปแบบ Dictionary ซึ่งประกอบไปด้วย Key ดังนี้

1) “input_ids” คือมี Value เป็น List ของค่าลักษณะของคู่คำถามและคำตอบ

2) “attention_mask” คือมี Value เป็น List ที่ใช้บอกแบบจำลองว่าโทเคนไหนต้องพิจารณาหรือไม่พิจารณา ใช้ใน Attention Mechanism โดย 1 คือพิจารณา และ 0 คือไม่พิจารณาหรือโทเคน < pad>

3) “offset_mapping” ใช้ในการหาตำแหน่งของช่วงคำตอบ (Answer Span)

4) “overflow_to_sample_mapping” ใช้ในการหาตำแหน่งของช่วงคำตอบ (Answer Span)

ยกตัวอย่างการใช้วิธีการของ Subword จากประโยคสองประโยค โดยประโยคกลางมีการตัดคำ ส่วนประโยคบนไม่มีการตัดคำ ถึงแม้การตัดคำของสองประโยคจะต่างกัน แต่ทั้งสองประโยคเป็นประโยคเดียวกัน และมี input_ids ต่างกัน ดังแสดงในรูปที่ 2.16 และ 2.17 [17]

```
1 print(tokenizer("ผมเป็นคนไทย")["input_ids"])
2 print(tokenizer(["ผม", "เป็น", "คนไทย"], is_split_into_words=True)["input_ids"])
[11, 162, 2939, 6762, 4949, 2617, 5901, 12]
[11, 162, 2939, 7439, 4344, 5901, 12]
```

รูปที่ 2.16 ค่า input_ids ของแต่ละคำในประโยค [17]

```
1 sent1 = tokenizer("ผมเป็นคนไทย")
2 sent2 = tokenizer(["ผม", "เป็น", "คนไทย"], is_split_into_words=True)
3 print(tokenizer.convert_ids_to_tokens(sent1["input_ids"]),
4       tokenizer.convert_ids_to_tokens(sent2["input_ids"]))
['[CLS]', 'ผม', '##น', '##เป็น', '##ค', '##น', '##ไทย', '[SEP]'] ['[CLS]', 'ผม', '##น', 'เป็น', 'คน', '##ไทย', '[SEP]']
```

รูปที่ 2.17 การแปลงกลับจาก input_ids ให้อยู่ในรูปแบบของประโยคที่ถูกโทเคน [17]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครื่องหมาย ## หมายถึง subword การที่ ['ผ', '##ม'] หมายความว่า 'ผ' กับ 'ม' ควรจะติดเป็นคำเดียวกัน [17]

และในปริยญาณิพนธ์นี้ ใช้รูปแบบของ Tokenization แบบ Subword Tokenization ที่ใช้อัลกอริทึม SentencePiece ซึ่งมีข้อดีที่ไม่ต้องการการตัดคำล่วงหน้า (Pretokenization) เหมือนตัวตัดคำย่อยอื่นๆ จึงเหมาะกับภาษาที่กฎการตัดคำไม่ตายตัวอย่างภาษาไทย แต่การนำ SentencePiece มาใช้นั้นมีปัญหาตรงที่ตัวตัดคำย่อยนี้มองว่า “ช่องว่างหรือเว้นวรรค” เป็นหนึ่งในตัวอักษรที่ถูกรวบเป็นคำย่อย ทำให้บางครั้ง “ช่องว่างหรือเว้นวรรค” ถูกทำให้หายไป ซึ่งแน่นอนว่าหาก “ช่องว่างหรือเว้นวรรค” หายไปแบบจำลองย่อมทำงานได้ไม่ดีในงานประเภทแยกคำ เพื่อแก้ปัญหานี้จึงมีการเพิ่มโทเคนพิเศษสำหรับช่องว่าง ‘_’ เพื่อให้ช่องว่างทุกช่องถูกรักษาไว้ แสดงดังรูปที่ 2.18 [18]

ข้อความตั้งต้น	วันนี้ ผมกินโมจิ
SentencePiece (XLMR)	[วันนี้, ผม, กิน, โม, จิ]
SentencePiece (WangchanBERTa)	[วันนี้, <_>, ผม, กิน, โม, จิ]

รูปที่ 2.18 ตัวอย่างการตัดคำโดยใช้อัลกอริทึม SentencePiece เพื่อรักษาช่องว่าง [18]

2.8 การวัดประสิทธิภาพของแบบจำลอง

ถือเป็นเครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย (Prediction) ที่ทำนายจากแบบจำลองที่สร้างขึ้น

2.8.1 Exact Match

Exact Match จะคำนวณอัตราที่ String ที่คาดคะว่าตรงกับคำตอบอ้างอิงทุกประการ เพียงใด โดยที่ Exact Match Score จะมีค่าเท่ากับ 1 เมื่อคำตอบที่คาดคะเนตรงกับคำตอบอ้างอิงทุกประการ และ Exact Match Score มีค่าเป็น 0 เมื่อคำตอบที่คาดคะเนไม่ตรงกับคำตอบอ้างอิง คะแนนดังกล่าวจะถูกนำมาหาค่าเฉลี่ย และได้เป็น Exact Match โดยรวม [19]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8.2 F1 – Score

ค่าที่แสดงประสิทธิภาพโดยการนำค่า Precision และ Recall มาคำนวณหาค่าเฉลี่ยหรือเรียกว่า Harmonic Mean ซึ่งค่าสูงๆถือว่า Model มีประสิทธิภาพดี ซึ่งค่าของ Precision และ Recall สามารถคำนวณได้จาก สิ่งที่คิด (แบบจำลองทำนาย) กับสิ่งที่เกิดขึ้นจริง ดังแสดงในรูปที่ 2.19 [20]

True Positive (TP) คือ สิ่งที่แบบจำลองทำนายว่า “จริง” และมีค่าเป็น “จริง”

True Negative (TN) คือ สิ่งที่แบบจำลองทำนายว่า “ไม่จริง” และมีค่าเป็น “ไม่จริง”

False Positive (FP) คือ สิ่งที่โปรแกรมทำนายว่า “จริง” และมีค่าเป็น “ไม่จริง”

False Negative (FN) คือ สิ่งที่โปรแกรมทำนายว่า “ไม่จริง” และมีค่าเป็น “จริง”

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

รูปที่ 2.19 ตาราง Confusion Matrix [20]

ค่า Precision คือ ค่าความแม่นยำ ดังสมการ 2.9

$$\frac{TPs}{TPs + FPs} \quad (2.9)$$

ค่า Recall คือ ความถูกต้องของการทำนายว่าจะเป็นจริงเทียบกับจำนวนครั้งของเหตุการณ์ทั้งทำนายและเกิดขึ้นว่าจริง ดังสมการที่ 2.10

$$\frac{TPs}{TPs + FNs} \quad (2.10)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นค่า F1 – Score มีสมการดังแสดงในสมการที่ 2.11

$$\frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}} \quad (2.11)$$

2.9 ฐานข้อมูล (Knowledge Base)

ฐานข้อมูล (Knowledge Base) [21] คือ กลุ่มของข้อมูลที่ถูกเก็บรวบรวมไว้อย่างเป็นระบบโดยมีความสัมพันธ์ซึ่งกันและกัน โดยไม่ได้บังคับว่าข้อมูลทั้งหมดนี้จะต้องเก็บไว้ในแฟ้มข้อมูลเดียวกันหรือแยกเก็บหลาย ๆ แฟ้มข้อมูล สามารถเก็บข้อมูลได้ทุกประเภท เช่น คำ, ตัวเลข, วิดีโอ และรูปภาพ โดยมีซอฟต์แวร์เข้ามาควบคุมกระบวนการใช้งาน การทำงาน หรือการประมวลผล

ระบบการจัดการฐานข้อมูล (Database Management System, DBMS) คือ โปรแกรมซอฟต์แวร์ฐานข้อมูลที่ครอบคลุม ทำหน้าที่เปรียบเสมือนสื่อกลางระหว่างโปรแกรมกับผู้ใช้งานช่วยให้ผู้ใช้สามารถดึงข้อมูลอัปเดตและจัดการหรือปรับแต่งข้อมูลได้ง่ายยิ่งขึ้น ระบบจัดการฐานข้อมูลที่นิยมใช้กันในปัจจุบัน [22] ได้แก่ Oracle, Microsoft SQL Server, MySQL เป็นต้น ปริมาณงานที่ผู้ใช้ MySQL เป็นระบบจัดการฐานข้อมูล

องค์ประกอบของระบบฐานข้อมูล [23] มีดังนี้

1) ข้อมูล (Data) คือ ข้อเท็จจริงที่เกิดขึ้นที่นำมาเก็บไว้เป็นฐานข้อมูลซึ่งข้อมูลอาจมีการซ้ำซ้อนกันได้ระหว่างแฟ้มต่าง ๆ แต่ต้องมีการซ้ำกันให้น้อยที่สุดหรือเท่าที่จำเป็น

2) ฮาร์ดแวร์ (Hardware) เป็นอุปกรณ์คอมพิวเตอร์ที่มีประสิทธิภาพที่พร้อมจะอำนวยความสะดวกในการบริหารระบบงานฐานข้อมูล

3) ซอฟต์แวร์ (Software) เป็นส่วนที่เชื่อมต่อระหว่างผู้ใช้กับฮาร์ดแวร์ ซึ่งเป็นโปรแกรมที่ทำหน้าที่ดูแลการสร้างการเรียกใช้ข้อมูลต่าง ๆ การปรับเปลี่ยน แก้ไข หรือกล่าวได้อีกอย่างว่าเป็น “ระบบการจัดการฐานข้อมูล (Database Management System, DBMS)”

4) บุคลากร (People) คือ บุคคลที่เกี่ยวข้องกับการใช้ฐานข้อมูล เช่น ผู้บริหารฐานข้อมูล (DataBase Administrator, DBA), นักวิเคราะห์และการออกแบบระบบ (System Analyst) และผู้ใช้ทั่วไป เป็นต้น

ประโยชน์ของฐานข้อมูล [24] มีดังนี้

1) ลดการซ้ำซ้อนของข้อมูลที่ต้องการเก็บ เนื่องจากข้อมูลบางชุดอาจปรากฏอยู่หลาย ๆ แห่ง เพราะมีผู้ใช้ชุดข้อมูลนี้หลายคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) รักษาความถูกต้องของข้อมูล เนื่องจากฐานข้อมูลมีเพียงข้อมูลเดียวหากเกิดกรณีที่มีข้อมูลเดียวกันปรากฏอยู่หลายแห่งในฐานข้อมูลข้อมูลเหล่านี้จะต้องถูกแก้ไขให้ตรงกันโดยการแก้ไขอัตโนมัติด้วยระบบจัดการฐานข้อมูล

3) มีการป้องกันและรักษาความปลอดภัยให้กับข้อมูลของผู้ใช้อย่างสะดวกซึ่งการป้องกันและรักษาความปลอดภัยนี้จะอนุญาตให้แค่ผู้เกี่ยวข้องเท่านั้นที่สามารถเข้าถึงได้

2.9.1 ระบบจัดการฐานข้อมูล MySQL

MySQL จัดเป็นระบบจัดการฐานข้อมูลเชิงสัมพันธ์ (Relational Database Management System, RDBMS) [25] สร้างขึ้นโดย Michael Widenius และ David Axmark ในปี 1995 MySQL เป็นซอฟต์แวร์ประเภท Open-Source Software ซึ่งเป็นที่นิยมกันมากในปัจจุบัน เพราะว่า MySQL เป็นฐานข้อมูลที่มีประสิทธิภาพสูงใช้ได้กับทุกอย่างตั้งแต่เว็บไซต์ที่มีขนาดเล็กไปจนถึงองค์กรขนาดใหญ่ อีกทั้งยังมีความรวดเร็ว เชื่อถือได้ และใช้งานได้ง่าย MySQL เป็นซอฟต์แวร์ที่ใช้ Structured Query Language (SQL) เป็นภาษาข้อมูลพื้นฐานและจัดเก็บข้อมูลในตาราง [26]

2.9.2 ภาษา Structured Query Language (SQL)

Structured Query Language (SQL) ถูกคิดค้นขึ้นในช่วงคริสต์ทศวรรษ 1970 เป็นภาษาโปรแกรมสำหรับการจัดเก็บและประมวลผลข้อมูลในฐานข้อมูลแบบเชิงสัมพันธ์ สามารถใช้ SQL ได้จากโปรแกรมต่าง ๆ ในการทำการดึงข้อมูล (Retrieve Data) จากฐานข้อมูลและเป็นมาตรฐานกลางที่ใช้ในระบบฐานข้อมูลต่าง ๆ โดยเป็นมาตรฐานของสถาบันมาตรฐานแห่งชาติอเมริกัน (American National Standard Institute, ANSI) [27]

2.10 Microservice

Microservice เป็นแนวคิดในการออกแบบและพัฒนาระบบซอฟต์แวร์ที่แบ่งแยกฟังก์ชันออกเป็นชิ้นย่อยที่เป็นอิสระต่อกัน มีวัตถุประสงค์เพื่อเพิ่มความยืดหยุ่นและความสามารถในการปรับปรุงแก้ไขระบบได้อย่างรวดเร็วขึ้นโดยไม่ต้องพัฒนาระบบทั้งหมดในครั้งเดียว ซึ่ง Microservice สามารถทำงานร่วมกันได้ผ่าน API

Microservice มีข้อดีหลายอย่าง [28] ได้แก่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) มีความยืดหยุ่น คือ แต่ละ Microservice มีอิสระต่อกันและสามารถพัฒนาแยกส่วนกันได้โดยที่ไม่กระทบกัน จึงทำให้การปรับปรุงและพัฒนาระบบซอฟต์แวร์นั้นรวดเร็วและมีความยืดหยุ่นสูง

2) ความเร็วและประสิทธิภาพ คือ แต่ละ Microservice มีการทำงานที่แยกกัน หากมีบริการหนึ่งพัง บริการอื่น ๆ ก็ยังสามารถทำงานต่อได้

3) ประหยัดทรัพยากร คือ แต่ละ Microservice สามารถปรับขนาดตามความต้องการได้ ทำให้ไม่ต้องใช้ทรัพยากรในการทำงานที่ไม่จำเป็น

สิ่งสำคัญสำหรับการสร้าง Microservice คือ การวางแผนออกแบบการบริการ ให้มีคุณสมบัติครบถ้วนเพื่อให้ได้รับประโยชน์จากการใช้งานบริการมากที่สุด ดังนั้นสิ่งที่มีประโยชน์สำหรับการสร้าง Microservice ได้แก่ API ซึ่งเป็นสิ่งที่แต่ละบริการสามารถติดต่อหรือเชื่อมต่อกันได้ ยิ่งระบบที่มีขนาดใหญ่ก็จะมีบริการต่าง ๆ จำนวนมาก ดังนั้นจึงต้องวางแผนการใช้งาน API ให้เหมาะสมเพื่อลดความซ้ำซ้อน และอีกหนึ่งสิ่งที่ต้องออกแบบ [28]

2.11 API (Application Programming Interface)

API ย่อมาจาก Application Programming Interface เป็นช่องทางเชื่อมต่อกันระหว่างโปรแกรม หรือระบบ ทั้งในเซิร์ฟเวอร์และซอฟต์แวร์ซึ่งในปัจจุบันจะหมายถึง ช่องทางในการเชื่อมต่อเพื่อแลกเปลี่ยนข้อมูลกันระหว่างแอปพลิเคชัน โดยการใช้งานแบ่งออกได้เป็น 3 รูปแบบ คือ

1) Private API หมายถึง API ที่ใช้งานเพื่อเชื่อมต่อกันระหว่างแอปพลิเคชันภายในองค์กรเท่านั้น ไม่สามารถเรียกใช้งานจากอินเทอร์เน็ตหรือแอปพลิเคชันภายนอกองค์กรได้

2) Partner API หมายถึง API ที่ใช้งานเพื่อเชื่อมต่อกันระหว่างคู่ค้า (Partner) ทางธุรกิจ ซึ่งจะจำกัดช่องทางการเชื่อมต่อ โดยจะเปิดให้เชื่อมต่อเฉพาะคู่ค้าแต่ละรายที่มีการตกลงกันไว้เท่านั้น ซึ่งอาจเป็นได้ทั้ง API ที่เรียกใช้งานได้ฟรีหรือมีการเรียกเก็บเงินตามจำนวนการใช้งาน (Call Volume) ก็ได้

3) Public API หมายถึง API ที่ผู้พัฒนาหรือเจ้าของ API นั้น ๆ เปิดให้ผู้พัฒนาแอปพลิเคชันอื่น (Developer) สามารถลงทะเบียนและเรียกใช้งานได้จากอินเทอร์เน็ต ซึ่งส่วนใหญ่จะเป็น API ที่สามารถเรียกใช้งานได้ฟรีหรือเรียกใช้ได้จำกัดจำนวน ซึ่งหากต้องการเรียกใช้แบบไม่จำกัดจำนวนอาจจะมีการคิดค่าบริการหรือต้องทำข้อตกลงเพื่อใช้งานในรูปแบบ Partner API แทน

2.11.1 REST API (RESTful Web Services)

การพัฒนา API ในปัจจุบันเราจะพัฒนาโดยทำงานในรูปแบบ REST API (Representational State Transfer) หมายถึง การสร้าง API ประเภท RESTful Web Services ซึ่งจัดเป็น Web Service รูปแบบหนึ่ง โดยยังทำงานอยู่บนพื้นฐานของโพรโทคอล HTTP หรือ HTTPS คือประกอบไปด้วยคำร้องขอ (Request) และการตอบรับ (Response) ตามรูปแบบ HTTP ที่แยกเป็นส่วน Header และ Body เพียงแต่เนื้อหาของส่วน Body ของคำร้องขอหรือคำตอบรับที่รับส่งกันอาจอยู่ในรูปแบบของ XML, SOAP, JSON

REST API จะมีการกำหนด Endpoint หรือ URL ให้แอปพลิเคชันหรือระบบอื่นเรียกใช้งานแบบสื่อความหมายในการเรียกใช้งาน เช่น API ที่มี URL Path ลงท้าย /users/john หมายถึง การสืบค้นหรือเรียกดูข้อมูลของผู้ใช้งานระบบที่ชื่อว่า John หรือ /create หมายถึงการเพิ่มข้อมูลใหม่ที่ฐานข้อมูล เป็นต้น

2.11.2 FastAPI และ Uvicorn

FastAPI เป็นอีกหนึ่งมอดูลของภาษา Python ที่ใช้สำหรับการพัฒนา API ซึ่ง FastAPI เน้นในเรื่องความเร็วในการพัฒนา โดยรองรับการทำงานตั้งแต่ Python เวอร์ชัน 3.6 เป็นต้นไป โดยการใช้งานมอดูล FastAPI ต้องมีการติดตั้ง ASGI (Gateway Interface) ด้วย ซึ่งทำหน้าที่เป็นตัวจัดการคำร้องขอที่เข้ามาสู่โปรแกรม Python ที่เป็น API คล้ายกับมอดูล Flask ซึ่งจะมี WSGI อยู่ภายใน แต่แตกต่างกันตรงที่ ASGI นั้นเราจะต้องทำการติดตั้งต่างหาก ซึ่ง ASGI ถูกพัฒนาเพิ่มเติมมาจาก WSGI เพื่อรองรับการทำงานแบบไม่พร้อมสัมพันธ์ (Asynchronous) อย่าง WebSocket และ HTTP/2

FastAPI ถูกพัฒนามาจาก Starlette ซึ่งเป็นหนึ่งใน ASGI Framework โดยจะทำงานร่วมกับ ASGI ในการจัดการกับคำร้องขอและการตอบกลับของ API ซึ่ง ASGI ในที่นี้คือ Uvicorn

2.12 โพรโทคอล HTTP และ HTTPS

2.12.1 โพรโทคอล HTTP

HTTP (Hyper Text Transfer Protocol) คือ โพรโทคอล หรือรูปแบบการสื่อสารบนระบบเครือข่ายอินเทอร์เน็ตซึ่งเป็นจุดกำเนิดของ WWW (World Wide Web) ถูกเชื่อมต่อกันแบบพิเศษที่ทำให้คอมพิวเตอร์เหล่านั้นสามารถเข้าถึงข้อมูลเนื้อหาที่เก็บไว้ภายในของแต่ละเครื่อง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ได้ โดยผ่านทางเบราว์เซอร์ เช่น Firefox, Google Chrome, Safari, Opera และ Microsoft Internet Explorer เพื่อการแลกเปลี่ยนข้อมูล ถ่ายโอนไฟล์ในรูปแบบสื่อผสม เช่น ข้อความ ภาพกราฟิก เสียง วิดีโอ และไฟล์มัลติมีเดียอื่น ๆ โดยการทำงานของ HTTP ทำงานอยู่ในระดับ Application Layer บนโพรโทคอล TCP/IP ใช้ URL เพื่อระบุ Server ปลายทางในการดึงและแลกเปลี่ยนข้อมูลซึ่งมีโครงสร้างเป็นตัวอักษรและตัวเลข ใช้สำหรับเป็นตัวเชื่อมโยง ข้อมูล Text และอื่นๆ ในรูปแบบ Plain Text เป็นข้อความธรรมดาไม่มีการเข้ารหัส ทำให้สามารถถูกดักจับและอ่านข้อมูลได้ง่าย จึงไม่ปลอดภัย โดยในการแลกเปลี่ยนข้อมูลระหว่างเว็บไคลเอนท์ (Web Client) และเว็บเซิร์ฟเวอร์ (Web Server) โดยการส่งจากเว็บไคลเอนท์ไปเซิร์ฟเวอร์จะเรียกว่า HTTP Request ส่วนข้อมูลที่เว็บเซิร์ฟเวอร์ตอบกลับมาที่เว็บไคลเอนท์จะเรียกว่า HTTP Response

2.12.2 โพรโทคอล HTTPS

HTTPS (Hypertext Transfer Protocol Secure, Hypertext Transfer Protocol over Secure Socket Layer หรือ HTTP over SSL) คือ โพรโทคอล หรือรูปแบบการสื่อสารบนระบบเครือข่ายอินเทอร์เน็ต ต่างกับ HTTP คือการเพิ่ม S หรือ Secure คือมีการใช้ SSL (Secure Socket Layer) และ TLS (Transport Layer Security) ในการเข้ารหัสข้อมูลระหว่างการส่ง ช่วยรักษาความถูกต้องของข้อมูลผู้ใช้และเก็บข้อมูลไว้เป็นความลับระหว่างคอมพิวเตอร์ของผู้ใช้กับเว็บไซต์ โดยมีความปลอดภัยและเป็นส่วนตัวระหว่างใช้งาน

HTTPS หรือ HTTP + SSL จุดที่สำคัญคือมีส่วนการพิสูจน์ตัวจริงเป็นการตรวจสอบเพื่อระบุตัวตนในการเข้าสู่เว็บไซต์ก่อนแลกเปลี่ยนข้อมูลโดยตรงกับทางเว็บเซิร์ฟเวอร์เป็นโพรโทคอลที่เข้ารหัสในการสื่อสาร โดยใช้ Asymmetric Algorithm เพื่อไม่ให้เกิดการโจรกรรมข้อมูลระหว่างกลาง อีกทั้งยังสามารถเข้ารหัสทั้ง 2 ทาง ระหว่าง เว็บไคลเอนท์-เว็บเซิร์ฟเวอร์ เพื่อป้องกันการปลอมแปลงข้อมูล และยังมีมั่นใจได้ว่าระหว่างการแลกเปลี่ยนข้อมูลจะไม่ถูกแกะ หรือ ปลอมแปลง เหมาะสำหรับธุรกิจที่มีข้อมูลเป็นความลับ เช่น ธนาคาร [29]

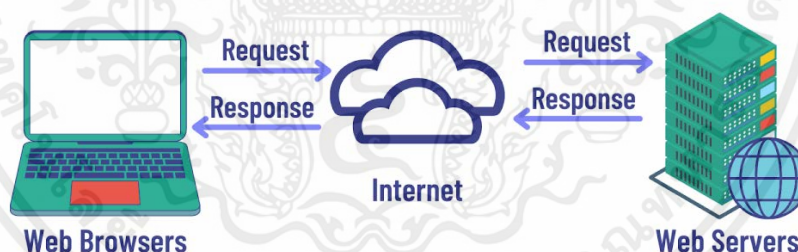
2.13 เว็บไซต์ (Website)

เว็บไซต์ (Website) หมายถึง หน้าเว็บเพจหลายหน้าที่จัดทำขึ้นเพื่อนำเสนอข้อมูลต่าง ๆ ผ่านทางคอมพิวเตอร์และอินเทอร์เน็ต ซึ่งเชื่อมโยงกันผ่านไฮเปอร์ลิงค์เพื่อให้สามารถเปิดไปยังหน้าเพจต่าง ๆ ได้อย่างง่ายดาย โดยเว็บไซต์ส่วนใหญ่ก็มีทั้งเว็บไซต์ที่เปิดให้เข้าชมได้ฟรีและบาง

เว็บไซต์ก็ต้องสมัครสมาชิกและเสียค่าบริการจึงจะเข้าใช้งานเว็บได้ ซึ่งข้อมูลในเว็บก็จะมีหลากหลายแบบขึ้นอยู่กับความต้องการนำเสนอของเจ้าของเว็บไซต์ [30]

กระบวนการทำงานของเว็บไซต์มีลักษณะการให้บริการในรูปแบบของไคลเอ็นท์เซิร์ฟเวอร์ (Client/Server) [31] โดยส่วนของไคลเอ็นท์หรือผู้ใช้บริการ หรือที่เรียกว่า เว็บเบราว์เซอร์ (Web browser) เช่น Internet Explorer, Google Chrome และ Firefox เป็นต้น ซึ่งโปรแกรมเหล่านี้จะทำหน้าที่เป็นผู้ร้องขอบริการจากผู้ให้บริการหรือเว็บเซิร์ฟเวอร์โดยที่การส่งคำร้องขอและการตอบกลับคำร้องขอนี้จะส่งผ่านกันบนโพรโทคอลสำหรับการสื่อสาร แสดงดังรูปที่ 2.20 [32]

ซึ่งการพัฒนาโปรแกรมในส่วนด้านหน้าที่มองเห็นได้ทันทีของหน้าเว็บไซต์นั้น เรียกว่า Front End Web Development เป็นการเขียนโปรแกรมให้ขึ้นไปแสดงผลได้บนเว็บเบราว์เซอร์ โดยทำตามมาตรฐาน W3C โดยต้องนำไฟล์ที่เขียนจากทางฝั่ง Front End Web Development ไปเก็บไว้บนเว็บเซิร์ฟเวอร์ซึ่งเป็นส่วนที่คอยป้อนไฟล์ของฝั่ง Front End ให้กับเว็บเบราว์เซอร์ กระบวนการนี้จะเป็นหน้าที่ของ Back End Web Development มีหน้าที่ในการจัดการข้อมูลที่ทาง Front End ต้องการและส่งไปให้รวมถึงการออกแบบฐานข้อมูลให้เก็บข้อมูลและดึงข้อมูลมาใช้ให้เร็วที่สุดได้โดยส่วนใหญ่เลือกส่งผ่าน API [33]



รูปที่ 2.20 แสดงการทำงานของการทำงานของการให้บริการเว็บไซต์ (Website) [32]

2.13.1 Web browser

Web browser คือ โปรแกรมคอมพิวเตอร์ที่ใช้ดูข้อมูลและโต้ตอบกลับข้อมูลทั้งหมดใน WWW. ซึ่งรวมถึงเว็บไซต์ เว็บเพจ วิดีโอ และรูปภาพ โดยส่วนประกอบของการพัฒนาหน้าเว็บเหล่านี้ถูกสร้างด้วยภาษาเฉพาะซึ่งทำตามมาตรฐาน World Wide Web Consortium (W3C) ได้แก่ HTML (HyperText Markup Language), CSS (Cascading Style Sheets) และ JavaScript [34]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.13.1.1 ภาษา HTML (Hyper Text Markup Language)

HTML ย่อมาจาก Hyper Text Markup Language คือภาษาคอมพิวเตอร์รูปแบบหนึ่งที่ใช้ในการแสดงผลของเอกสารหน้าเว็บ (Web page) ซึ่งถือว่าภาษา HTML เป็นโครงสร้างของเว็บไซต์ ถูกพัฒนาและกำหนดมาตรฐานโดยองค์กร World Wide Web Consortium (W3C) [35] การเรียกใช้งานหรือทดสอบการทำงานของเอกสาร HTML จะแสดงผลผ่านเว็บเบราว์เซอร์ เช่น IE Microsoft Internet Explorer (IE), Mozilla Firefox, Safari, Opera, และ Netscape Navigator เป็นต้น HTML มีรุ่นต่าง ๆ ออกมาอย่างต่อเนื่องตั้งแต่ HTML Level 1, HTML 2.0, HTML 3.0, HTML 3.2, HTML 4.0, HTML 4.01 และ HTML5 ในปัจจุบันนี้ผู้จัดทำได้ใช้ HTML5 สำหรับการสร้างเว็บไซต์ [36]

HTML5 เป็นภาษาที่ถูกพัฒนาต่อมาจากภาษา HTML และพัฒนาขึ้นมาโดย WHATWG (The Web Hypertext Application Technology Working Group) โดยได้มีการปรับปรุงเพิ่มค่าลักษณะหลาย ๆ อย่างเข้ามาเพื่อให้ผู้พัฒนาสามารถใช้งานได้ง่ายมากยิ่งขึ้น ข้อดีของ HTML5 [37] เช่น เว็บไซต์ที่สร้างจากภาษา HTML5 สามารถแสดงผลได้กับทุกเว็บเบราว์เซอร์, มีความเป็นอิสระสูงและมีการจัดการข้อผิดพลาดที่ดีขึ้น เป็นต้น

2.13.1.2 ภาษา CSS (Cascading Style Sheets)

ภาษา CSS ย่อมาจาก Cascading Style Sheets เรียกโดยย่อว่า "สไตล์ชีต" คือภาษาที่ใช้ในการตกแต่งเว็บไซต์หรือปรับแต่งเอกสาร HTML ให้มีรูปแบบที่สวยงามมากขึ้น เช่น ปรับแต่งเรื่องรูปแบบตัวอักษร, ขนาดของตัวอักษร, สีเส้นต่าง ๆ, รูปภาพ และอื่น ๆ [38] ซึ่ง CSS มีลักษณะเป็นภาษาที่มีรูปแบบในการเขียน Syntax แบบเฉพาะและได้ถูกกำหนดมาตรฐานโดย World Wide Web Consortium (W3C) [39] CSS มีทั้งหมด 3 รุ่น ได้แก่ CSS1, CSS2, CSS3

2.13.1.3 ภาษา JavaScript

JavaScript คือ ภาษาคอมพิวเตอร์ภาษาหนึ่งสำหรับการเขียนโปรแกรมบนระบบอินเทอร์เน็ต JavaScript เป็นภาษาสคริปต์เชิงวัตถุซึ่งในการสร้างและพัฒนาเว็บไซต์จะใช้ร่วมกับ HTML เพื่อให้เว็บไซต์มีลูกเล่นต่าง ๆ ช่วยให้มีความน่าสนใจมากยิ่งขึ้น [40]

2.13.1.4 Cloudinary

Cloudinary คือ Software-as-a-Service (SaaS) หรือ บริการซอฟต์แวร์บนอินเทอร์เน็ตที่ทำให้สามารถนำมาใช้งานได้ทุกที่ โดยหน้าที่หลักของ Cloudinary คือ การจัดเก็บไฟล์รูปภาพและวิดีโอของเว็บไซต์ หรือโปรเจกต์ต่าง ๆ แทนที่จะเก็บไว้ใน Server หรือเก็บลงใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โพลเดอร์ และสามารถปรับแต่งขนาด, สีและเอฟเฟคของรูปภาพได้ง่าย ๆ ผ่าน Cloudinary โดยตรง [41]

2.13.2 Web Server

Web server คือ ซอฟต์แวร์หรือโปรแกรมสำหรับเซิร์ฟเวอร์รวมถึงฮาร์ดแวร์ซึ่งก็คือ เครื่องคอมพิวเตอร์ [42] โดยหน้าที่หลักของเว็บเซิร์ฟเวอร์ คือการจัดเก็บ, รับคำสั่งจากการร้องขอของฝั่งไคลเอ็นท์โดยผ่านทางเว็บเบราว์เซอร์และมีการประมวลผลการทำงานที่มาจากการร้องขอดังกล่าวแล้วส่งคำสั่งร่อนกลับไปยังเครื่องของไคลเอ็นท์ที่ร้องขอโดยส่งคำสั่งร่อนเหล่านั้นผ่านทางโพรโทคอลสำหรับการสื่อสารระหว่างไคลเอ็นท์และเซิร์ฟเวอร์ เช่น HTTP, WebSocket ทั่วไปแล้ว Web server มีการใช้งานร่วมกับภาษาโปรแกรมหลากหลาย เช่น PHP, Python และ Node.js ซึ่งสามารถเลือกได้ตามลักษณะของการทำงานและความต้องการของโครงการ ปรินูญา นิพนธ์นี้ใช้โพรโทคอล WebSocket และใช้ Node.js เป็นภาษาโปรแกรม

2.13.2.1 Node.js

Node.js คือ Runtime Environment สำหรับการใช้ภาษา JavaScript นอกเว็บเบราว์เซอร์บนฝั่ง Server-Side ซึ่ง Node.js เป็น Open Source และ Library ที่ใช้สำหรับพัฒนาเว็บแอปพลิเคชันต่าง ๆ ด้วยภาษา JavaScript โดยใช้เทคโนโลยี V8 JavaScript Engine ของ Google ซึ่งเป็น Engine ที่ดำเนินการ JavaScript ได้เร็วและมีประสิทธิภาพสูง นอกจากนี้ Node.js ยังเป็นแพลตฟอร์มที่มีความยืดหยุ่นสูงในการพัฒนาเว็บแอปพลิเคชัน (Web Application) และ Network Application ต่าง ๆ ด้วยการใช้งานแบบ Asynchronous I/O รวมถึงสามารถเข้าถึงฐานข้อมูลได้แบบ Non-Blocking I/O อีกด้วย

อีกทั้ง Node.js ยังมี Framework เพื่อช่วยให้ผู้พัฒนาโปรแกรมสามารถพัฒนาแอปพลิเคชัน Node.js ได้อย่างรวดเร็วและสะดวกสบายมากขึ้น ตัวอย่างของ Framework บน Node.js ที่ได้รับความนิยมและถูกนำมาใช้งานอย่างแพร่หลายได้แก่ Express, Koa, Hapi, Meteor และ NestJS ซึ่งทั้งหมดนี้มีความสามารถและพื้นฐานที่แตกต่างกันไปตามแต่ละ Framework ในการใช้งาน

2.13.2.1.1 Express Framework

Framework คือ ชุดคำสั่ง เครื่องมือ หรือโครงสร้างอย่างใดอย่างหนึ่งที่ถูกสร้างไว้เพื่ออำนวยความสะดวกให้กับนักพัฒนาโปรแกรม [43] จึงกล่าวได้ว่า Framework นั้นเป็นเครื่องมือที่มีประโยชน์อย่างมากสำหรับนักพัฒนาโปรแกรม ซึ่งประโยชน์ของ Framework มีดังนี้ [43]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1) ประหยัดเวลาและแรงงานของนักพัฒนาโปรแกรม
 2) Framework มีการจัดเตรียมชุดคำสั่งพื้นฐานต่าง ๆ เพื่อให้
 นำไปต่อยอดได้ง่ายมากขึ้น

3) มีการสร้างชุดคำสั่งที่เป็นระเบียบและสามารถปรับเปลี่ยนได้
 ปริญญาโทฉบับนี้ใช้ Express Framework เป็น Framework
 สำหรับ Node.js ที่ช่วยสร้างแอปพลิเคชันแบบ Server-side ได้อย่างง่ายดายและมีความรวดเร็วมาก
 ยิ่งขึ้น โดยสามารถใช้งานมอดูลต่าง ๆ ได้ เช่น middleware, routing, template engines และ
 อื่น ๆ นอกจากนี้ Express ยังมีความสามารถในการทำงานร่วมกับหลาย ๆ Framework และ
 Library ต่าง ๆ เช่น body-parser, helmet, morgan, และ Passport ทำให้นักพัฒนาสามารถ
 สร้างแอปพลิเคชันที่มีความปลอดภัยและมีความสามารถหลากหลายด้าน ดังนั้น Express จึงเป็น
 Framework ที่ดีและเป็นนิยมใช้กันอย่างมากใน Node.js สำหรับการสร้างเว็บแอปพลิเคชัน

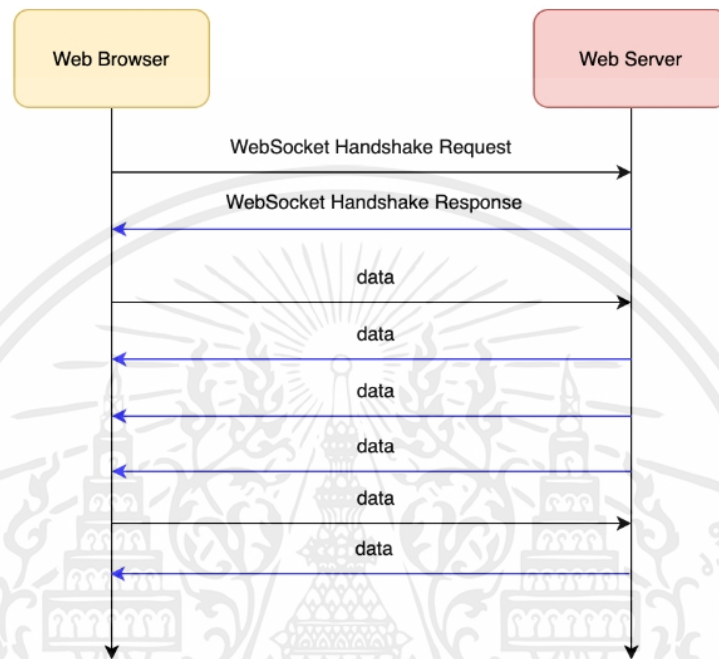
2.13.2.2 Socket.io และ WebSocket

Socket.io เป็น Library ของ JavaScript ที่ใช้งานกับ Node.js ซึ่งช่วยใ้
 นักพัฒนาโปรแกรมสามารถที่จะสร้างแอปพลิเคชันได้แบบ Real Time ได้ง่ายยิ่งขึ้น โดยการใช้งาน
 Socket.io จะเป็นการเชื่อมต่อแบบคู่กันระหว่างเว็บเบราว์เซอร์และเว็บเซิร์ฟเวอร์ซึ่ง Socket.io
 เป็น Library ที่ช่วยให้การใช้งานของ WebSocket ที่เป็นโพรโทคอลสำหรับการสื่อสารมีความ
 สะดวกและยืดหยุ่นมากขึ้น

WebSocket คือ โพรโทคอลในการสร้างการเชื่อมต่อสื่อสารระหว่าง
 เว็บเบราว์เซอร์และเว็บเซิร์ฟเวอร์ [44] เป็นการสื่อสารแบบ Full Duplex หรือ Bidirectional
 Communication และมีการเปิดการใช้งานการสื่อสารกันแบบ Real Time ซึ่งเหมาะกับงาน ดังนี้
 แพลตฟอร์มแลกเปลี่ยน, แอปพลิเคชันเกม และแชตบอต

การทำงานของ WebSocket จะเกิดเมื่อเว็บเบราว์เซอร์ต้องการร้องขอ
 ข้อมูลจากเว็บเซิร์ฟเวอร์ผ่าน WebSocket โดยเว็บเบราว์เซอร์จะเริ่มต้นด้วยการส่ง HTTP request
 เพื่อทำการแลกเปลี่ยนข้อมูลแบบ Handshake หรือ WebSocket Handshake Request จากนั้น
 เว็บเซิร์ฟเวอร์ตอบกลับเป็น HTTP response หรือ WebSocket Handshake Response แต่ยังไม่
 มีข้อมูลใด ๆ กลับไปยังเว็บเบราว์เซอร์พร้อมกับ HTTP header ที่ระบุว่า WebSocket ได้ถูก
 เปิดขึ้นแล้ว จากนั้นจึงเริ่มกระบวนการแลกเปลี่ยนข้อมูลกันระหว่างเว็บเบราว์เซอร์และเว็บ
 เซิร์ฟเวอร์โดยไม่ต้องสร้างการเชื่อมต่อใหม่ทุกครั้งที่มีการส่งข้อมูลซึ่งทำให้การสื่อสารด้วย

WebSocket มีความเร็วและประสิทธิภาพสูงกว่าการสื่อสารด้วย HTTP request/response ธรรมดา แสดงดังรูปที่ 2.21 [45]



รูปที่ 2.21 การทำงานของ WebSocket [45]

2.14 Text-to-Speech (TTS)

Text-to-Speech (TTS) เป็นเทคโนโลยีที่ช่วยแปลงจากข้อความให้กลายเป็นเสียงพูดได้อัตโนมัติ โดยที่ไม่จำเป็นต้องมีคนพูดจริง คือ สามารถแปลงข้อความจากเครื่องคอมพิวเตอร์ให้เป็นเสียงพูดได้โดยอัตโนมัติการทำงานของ Text-to-Speech มีหลายวิธี แต่วิธีที่ได้รับความนิยมสูงสุดคือการใช้แบบจำลองการพูดที่ถูกสร้างขึ้นด้วยเทคโนโลยีการเรียนรู้เชิงลึกเพราะแบบจำลองเหล่านี้จะมีการเรียนรู้จากเสียงพูดจริงแล้วจึงสร้างเสียงพูดขึ้นมาใหม่ได้เอง

การทำงานของ Text-to-speech จะเริ่มต้นด้วยการนำข้อความเข้ามาเป็นข้อมูล จากนั้นแบบจำลองจะทำการแปลงข้อความให้เป็นเสียงพูด โดยการนำข้อมูลเสียงที่ถูกสร้างขึ้นมาผ่านการปรับแต่งและประมวลผลเพื่อให้เสียงดูเป็นเสียงพูดของมนุษย์มากที่สุด และสุดท้ายจะมีการ

ส่งออกเสียงพูดไปยังผู้ใช้งานซึ่งอาจเป็นผ่านทางเว็บไซต์ แอปพลิเคชัน หรืออุปกรณ์อื่น ๆ ที่มีการรองรับ Text-to-speech

การใช้งาน Text-to-Speech มีประโยชน์อย่างมาก เช่น ช่วยเพิ่มความสะดวกสบายในการอ่าน ช่วยให้ผู้พิการทางสื่อสารสามารถเข้าถึงเนื้อหาที่เขียนไว้หรือเว็บไซต์ได้ง่ายขึ้น

2.15 Speech-to-Text (STT)

Speech-to-Text (STT) คือ เทคโนโลยีการแปลงเสียงพูดให้กลายเป็นข้อความ โดยจะใช้แบบจำลองการเรียนรู้ของเครื่อง (Machine Learning) และการเรียนรู้เชิงลึก ที่เรียนรู้และปรับปรุงความสามารถจากข้อมูลเสียงที่ได้รับการเก็บตัวอย่างมา โดยปกติแล้ว Speech-to-Text จะเรียกใช้งานผ่าน API ที่ถูกสร้างขึ้นโดยบริษัทหรือผู้ให้บริการเท่านั้น

เมื่อมีการใช้งาน Speech-to-Text สามารถพูดคำพูดเข้าไปในอุปกรณ์ที่มีมอดูล Speech-to-Text เช่น สมาร์ทโฟน หรือโทรศัพท์ที่รองรับ Speech-to-Text (STT) แล้วมอดูลจะรับเสียงที่เข้ามาแล้วแปลงเสียงเหล่านั้นให้กลายเป็นข้อความ จากนั้นก็จะส่งข้อความที่ได้กลับไปยังอุปกรณ์ของผู้ใช้งานเพื่อให้ผู้ใช้งานสามารถอ่านและเข้าใจข้อความนั้นได้

Speech-to-Text มีประโยชน์อย่างมากในการช่วยเหลือผู้พิการทางการได้ยินและการพูด โดยช่วยเพิ่มโอกาสในการเข้าถึงสื่อต่าง ๆ และความสะดวกในการควบคุมอุปกรณ์ด้วยเสียง นอกจากนี้ยังมีการนำ Speech-to-Text ไปใช้ในงานด้านธุรกิจ เช่น การบันทึกข้อความในการประชุม การเข้าใจข้อความที่ไม่เป็นภาษาต้นทาง หรือการค้นหาข้อมูลในเว็บไซต์ที่สามารถใช้เสียงเป็นตัวกำหนดการค้นหาได้

2.16 กราฟิกเคลื่อนไหว (Motion Graphic)

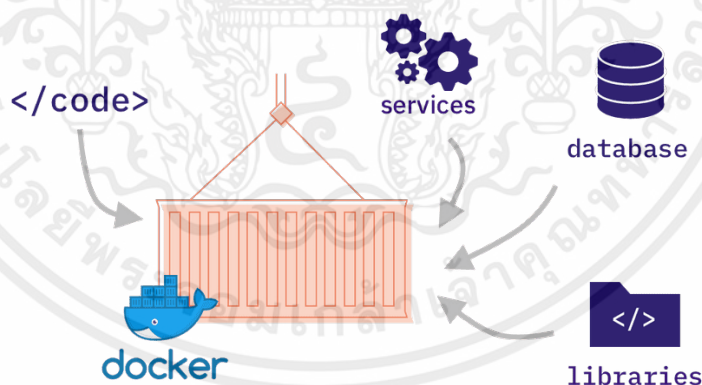
กราฟิกเคลื่อนไหว (Motion Graphic) เป็นรูปแบบงานกราฟิกที่สร้างจากภาพวาด 2 มิติโดยมีระบบการทำงานในรูปแบบ Animation หรือสร้าง Digital footage ที่สร้างภาพเคลื่อนไหวให้ปรากฏบนจอภาพ ซึ่งอาจจะประกอบด้วยการใส่เสียงและสื่อผสมต่าง ๆ โดยปัจจุบันนิยมและพัฒนายิ่งขึ้นทั้งด้านของสื่อและศิลปะร่วมสมัยอย่างหลากหลายเนื่องจากสามารถอธิบายและทำความเข้าใจได้ง่ายเมื่อถูกสื่อออกไป โปรแกรมที่ใช้ออกแบบ กราฟิกเคลื่อนไหว (Motion Graphic) เช่น Adobe Photoshop, Adobe ImageReady, After Effect,

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Clip-Studio Paint และ Adobe Premiere Pro ปัญญาประดิษฐ์ใช้ Clip-Studio Paint สำหรับวาดและทำ Animation และใช้ Adobe Premiere Pro สำหรับลบภาพพื้นหลัง

2.17 Docker

Docker คือเครื่องมือแบบ Open Source หรือแพลตฟอร์มซอฟต์แวร์ที่ช่วยจำลองสภาพแวดล้อม (Environment) ในการรัน Service หรือ Server [46] ช่วยให้สามารถสร้าง ทดสอบ และติดตั้งแอปพลิเคชันได้อย่างรวดเร็ว โดย Docker จะบรรจุซอฟต์แวร์ลงไปเป็นหน่วยที่เป็นมาตรฐาน เรียกว่า Container ซึ่งจะมีทุกสิ่งที่ซอฟต์แวร์ต้องใช้ในการเรียกใช้งาน รวมทั้ง Library เครื่องมือสำหรับระบบ โค้ด และรันไทม์ สามารถติดตั้งใช้จริงและปรับขนาดแอปพลิเคชันให้เหมาะกับทุกสภาพแวดล้อม (Deploy) [47] หรืออธิบายอีกนัยได้ว่า การใช้ “Docker” เข้ามาช่วยในการอัดซอฟต์แวร์หนึ่งให้อยู่ในรูปแบบของ Container ตัว Container นี้สามารถนำไปรันบนเครื่องไหนสภาพแวดล้อมไหนก็ได้ ซึ่งแอปพลิเคชันดังกล่าวจะสามารถทำงานได้ปกติ ไม่ว่าจะนำไปรันบนเครื่อง Development หรือบน Production Server ไหนก็ตาม [48] ตัวอย่างภาพรวมการใช้ Docker แสดงดังรูปที่ 2.22 [49]



รูปที่ 2.22 ตัวอย่างภาพรวมการใช้ Docker [49]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

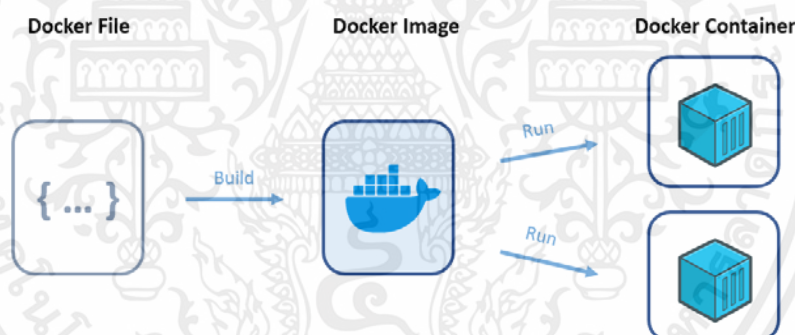
2.17.1 องค์ประกอบพื้นฐานของ Docker

Docker ทำงานด้วย 3 ขั้นตอนได้แก่ Build, Ship และ Run ซึ่งจะใช้อะไรประกอบพื้นฐาน 3 อย่างดังนี้ โดยสำหรับองค์ประกอบพื้นฐานและการทำงานของ Docker แสดงดังรูปที่ 2.23 [46]

1) Dockerfile (Build) คือไฟล์ที่นำเอา Source Code และ Config ที่รวบรวมคำสั่งทุกอย่างที่ใช้ในการรันแอปพลิเคชันไว้สำหรับสร้างเป็น Docker Image

2) Docker Image (Ship) คือแม่แบบที่ใช้ในการสร้างเป็น Docker Container ซึ่งประกอบไปด้วยแอปพลิเคชันที่จะทำงานเมื่อมีการเรียกใช้งานจาก Docker Container นั้น ๆ รวมทั้งการตั้งค่าจำลองสภาพแวดล้อมที่จำเป็นสำหรับการทำงานไว้

3) Docker Container (Run) เป็นที่บรรจุรวมของแอปพลิเคชันสภาพแวดล้อมที่จำเป็นต่อการทำงาน และองค์ประกอบต่าง ๆ ที่จำเป็นต่อการทำงาน ซึ่งสามารถสร้างจาก Docker Image ผ่านการกำหนดโครงสร้างที่ Dockerfile



รูปที่ 2.23 องค์ประกอบพื้นฐานและการทำงานของ Docker [46]

2.17.2 ข้อแตกต่างของ Docker เมื่อเทียบกับ Virtual Machine

Docker มีส่วนที่คล้ายกับ Virtual Machine (VM) คือ สามารถแยกเนื้อหาออกจากซอฟต์แวร์อื่น ๆ ที่กำลังทำงานอยู่ในเครื่องได้ Docker มีส่วนที่แตกต่างจาก Virtual Machine ในหลาย ๆ ด้าน ในขณะที่ Virtual Machine เป็นโฮสต์ของระบบปฏิบัติการทั้งหมด แต่ Docker Container นั้นไม่ใช่ เพราะ Docker จะไม่ทำงานบนไฮเปอร์ไวเซอร์ Type-1 ที่เข้ามาแทนที่ระบบปฏิบัติการหลักของคอมพิวเตอร์ เนื่องจาก Virtual Machine ต้องทำงานบน Server หลาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัว แต่จุดเด่นของ Docker ก็คือการใช้ทรัพยากรที่น้อยกว่าการทำงานแบบ Virtual Machines ดังนั้น Docker จึงถูกสร้างมาเพื่อแก้ไขข้อจำกัดของ Virtualization แต่ Docker จะเก็บเฉพาะสิ่งที่จำเป็นในการเรียกใช้บริการซอฟต์แวร์เดี่ยว ๆ เช่น ฐานข้อมูล หรือ เว็บเซิร์ฟเวอร์ ซึ่งรวมถึงบริการของตัว Docker เอง ตลอดจน Dependency ต่าง ๆ ที่จำเป็นต้องเรียกใช้ เช่น Library ของซอฟต์แวร์ ดังนั้น จึงไม่มีการบริการของระบบปฏิบัติการหลัก (Underlying Operating System Services) ใน Docker จึงเป็นสาเหตุที่ทำให้ Docker มีน้ำหนักเบา เนื่องจากต้องการพื้นที่ในการเก็บข้อมูลและหน่วยความจำที่น้อยกว่าในการทำงาน [50]

2.18 Google Cloud Platform (GCP)

Google Cloud Platform (GCP) หรือ Google Cloud เป็น Public Cloud ของ Google ที่รู้จักกันดี ไม่ว่าจะเป็น Google Search engine, Google Map, Google Drive โดยปกติแล้ว Cloud นั้นจะมีบริการหลายอย่าง [51] GCP ก็เช่นกัน โดย GCP มีบริการต่าง ๆ มากมาย ไม่ว่าจะเป็น บริการในกลุ่มของ AI และ การเรียนรู้ของเครื่อง, API Management, Compute, Containers, Data Analytics, Databases และ Storage เป็นต้น [52] โดยสำหรับปริญญาโทนี้ ผู้จัดทำได้ใช้บริการ Cloud Build จากบริการกลุ่ม Containers ซึ่งเป็น Service ที่เป็น Serverless CI/CD Platform สามารถจัดการ Testing, Building และ Deploying ได้ในตัวเอง, Cloud Registry ซึ่งเป็นการบริการที่ให้บริการที่จัดเก็บและจัดการ Container Image มีลักษณะคล้ายคลึงกับ Docker Hub โดยสามารถจัดเก็บได้ตาม Tag หรือ Versioning ของ Container Image และสามารถจัดเรื่องความเป็นส่วนตัวของ Container Image ได้ ว่าต้องการให้เป็นส่วนบุคคล หรือ สาธารณะเพื่อจัดการความปลอดภัย สามารถจัดการอนุญาตให้ใครสามารถเข้าถึง Container Image นั้น ๆ ได้, Cloud Run จากบริการกลุ่ม Compute โดยเป็น Compute Platform แบบ Serverless ที่สามารถสเกลขึ้นลงโดยอัตโนมัติได้ตามจำนวนการสื่อสารที่เข้ามา โดย Cloud Run จะทำหน้าที่ในการรัน Container Image จาก Cloud Registry และ Cloud SQL จากบริการกลุ่ม Databases ซึ่งเป็นบริการฐานข้อมูลแบบ Relational Database บน Google Cloud ที่เป็นการรันซอฟต์แวร์ฐานข้อมูลแบบ Fully Managed ซึ่ง Google จะดูแลระบบทุกอย่างให้ โดยรองรับทั้ง MySQL, PostgreSQL และ SQL Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.19 Heroku

Heroku เป็นบริการ Cloud แบบ Platform as a Service (PaaS) เป็นการพัฒนาของบริษัท Salesforce [53] ที่ให้บริการสำหรับนักพัฒนาซอฟต์แวร์ ซึ่งให้บริการในการฝากโค้ดและการทำงานของเว็บแอปพลิเคชันบน Cloud โดย Heroku จะทำหน้าที่จัดการเรื่องเซิร์ฟเวอร์และโครงสร้างพื้นฐาน ทำให้นักพัฒนาซอฟต์แวร์สามารถโฟกัสเฉพาะส่วนของการ Build, Deploy, Monitor App [53] และช่วยให้ผู้พัฒนาสามารถเน้นไปที่การพัฒนาแอปพลิเคชันได้มากกว่าการจัดการเซิร์ฟเวอร์อีกทั้งยังมีการ Add-ons สำหรับบริการอื่น ๆ เช่น Redis, PostgreSQL, MongoDB สามารถเลือกใช้งานได้ทั้งแบบฟรีและเสียเงิน [54]

Heroku เหมาะสำหรับนักพัฒนาที่ต้องการทดลองเขียนเว็บแอปพลิเคชันของตนและต้องการความยืดหยุ่นในการจัดการโครงสร้างพื้นฐานและต้องการส่วนติดต่อกับฐานข้อมูลแบบ Managed Service ซึ่ง Heroku ได้ทำการจัดการเอาไว้แล้ว และใช้งานง่าย [54]

ข้อดีของ Heroku คือ ไม่ต้องกังวลเรื่องการจัดการเซิร์ฟเวอร์สามารถทำ Scaling ได้ ง่ายตาย มีเครื่องมือในการจัดการแอปพลิเคชันและติดตามสถานะการทำงานได้อย่างสะดวก และยังสามารถใช้งานได้ฟรีหรือจ่ายค่าบริการตามการใช้งานจริงได้อีกด้วย

2.20 Git

Git คือ ระบบควบคุมเวอร์ชัน (Version Control System) ที่ใช้ในการจัดการ ตรวจสอบ แก้ไข และติดตามการเปลี่ยนแปลงของ Source Code [55] โดย Git จะเก็บประวัติของแต่ละไฟล์ที่ถูกสร้าง ลบ หรือแก้ไขโดยใคร เมื่อไหร่ ไว้ทั้งหมด ซึ่งช่วยให้นักพัฒนาสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพ Git มีให้เลือกใช้หลากหลายรูปแบบ เช่น Github, Gitlab หรือ Git ของตัวเองก็ได้

Git เหมาะสำหรับนักพัฒนาซอฟต์แวร์ทุกรูปแบบ ทั้งนักพัฒนาที่ยังคงเป็นมือใหม่หรือระดับมืออาชีพ โดย Git ช่วยให้นักพัฒนาสามารถทำงานร่วมกันได้อย่างมีประสิทธิภาพและสามารถตรวจสอบและแก้ไขข้อผิดพลาดได้อย่างรวดเร็ว

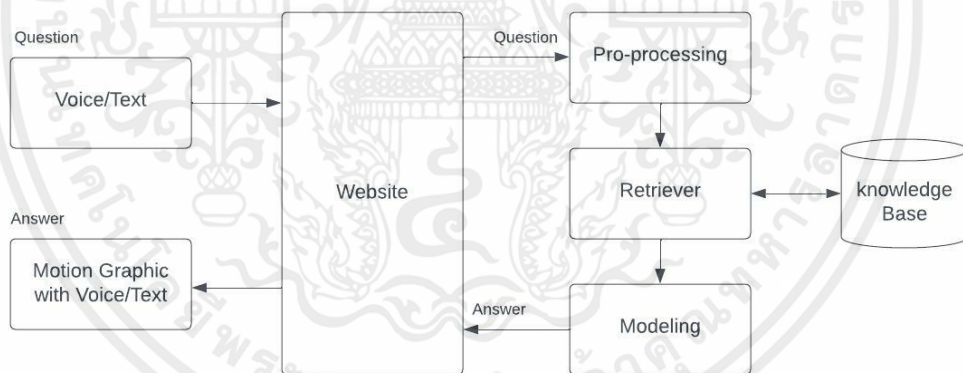
บทที่ 3

การออกแบบและการจัดทำปฏิญญานิพนธ์

3.1 การออกแบบ

3.1.1 การออกแบบระบบรวม

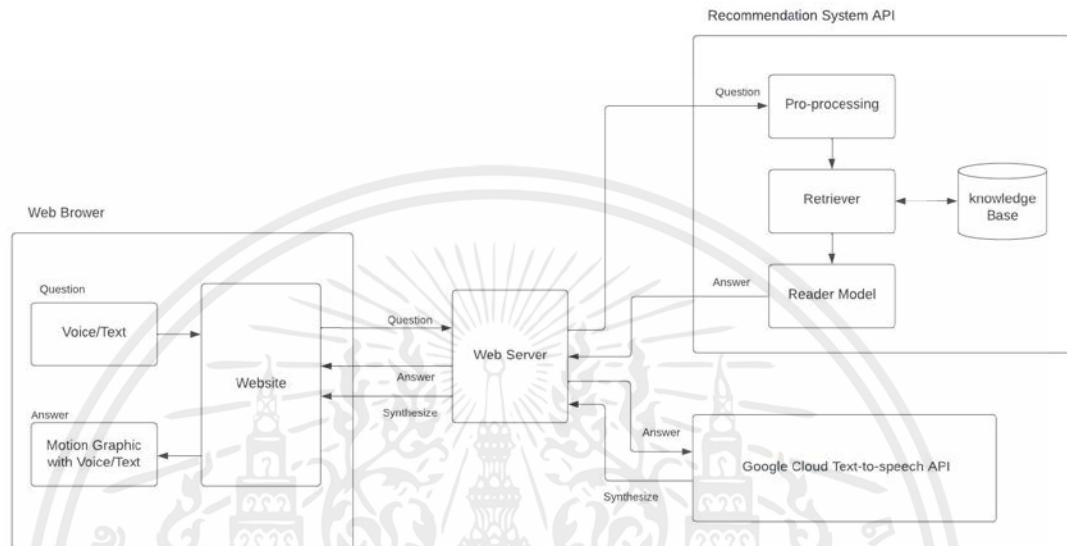
ศึกษาและพัฒนาระบบแนะนำข้อมูล (Recommendation System) ซึ่งจะแนะนำและตอบคำถามด้วยวิธีการประมวลผลภาษาธรรมชาติ (Natural Language Processing) โดยสร้างเป็นแบบจำลองสำหรับตอบคำถาม (Modeling) ในเรื่องของหลักสูตรวิศวกรรมโทรคมนาคม สาขาวิศวกรรมโทรคมนาคมและโครงข่าย คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง หลักสูตรปริญญาตรี ซึ่งจะทำการสร้างเป็นส่วนขยายสำหรับนำไปใช้บนเว็บไซต์ สามารถโต้ตอบผ่านข้อความหรือเสียงพูดและแสดงผลเป็นกราฟิกเคลื่อนไหวออกมาด้วย ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 บล็อกไดอะแกรมการศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

โดยในปฏิญญานิพนธ์นี้จะออกแบบให้ระบบในส่วน of Pre-processing, Retriever, Knowledge Base และ Model สร้างเป็น RESTfulAPI และจะทำการสร้างเว็บไซต์เพื่อนำ API

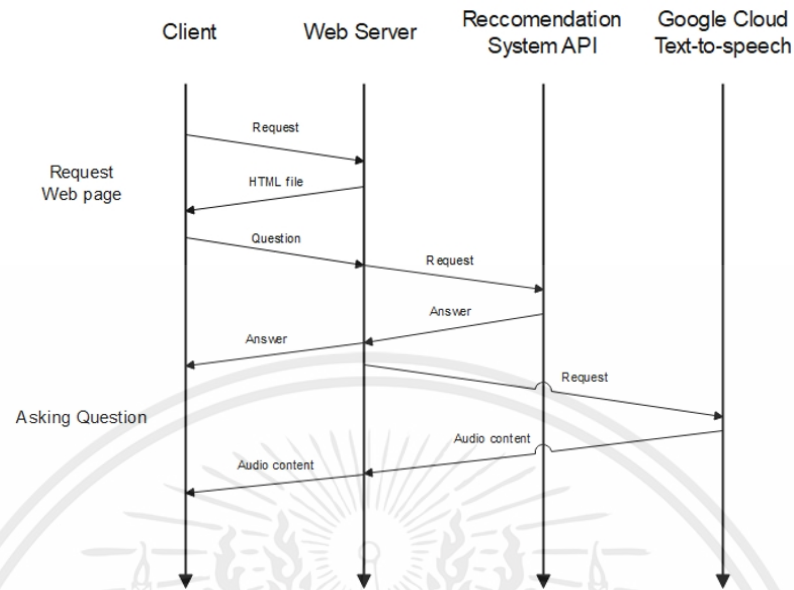
ของระบบนี้ไปใช้งานต่อไป นอกจากนี้ จะมีการนำ Google Cloud Text-to-speech API มาใช้งานในส่วน of เว็บไซต์อีกด้วย ดังแสดงในรูปที่ 3.2



รูปที่ 3.2 ภาพรวมการทำงานทั้งหมดของปัญญาประดิษฐ์

โดยขั้นตอนการใช้งานจะเริ่มจากผู้ร้องขอ (Request) หน้าเว็บไซต์จากเว็บเซิร์ฟเวอร์ต่อมาเมื่อผู้ใช้งานได้มีการถามคำถาม คำถามนี้จะถูกส่งมาที่เว็บเซิร์ฟเวอร์ก่อนจะส่งไปให้ Recommendation System API ตอบคำถามและส่งคำตอบกลับมา จากนั้นคำตอบนี้จะตอบกลับไปให้ผู้ใช้งานพร้อมกับถูกส่งต่อไปที่ Google Cloud Text-to-speech API เพื่อทำกระบวนการ Text-to-speech และจะได้ไฟล์เสียงกลับมาที่เว็บเซิร์ฟเวอร์ก่อนจะถูกส่งต่อไปให้ผู้ใช้งานอีกที ดังแสดงในรูปที่ 3.3

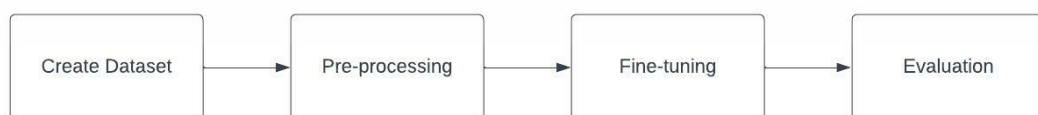
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 Network Timing Diagram

3.1.2 การออกแบบและสร้าง Reader Model

การสร้างแบบจำลองตัวอ่านสำหรับระบบถามตอบ โดยใช้ WangchanBERTa Pre-trained Model มีขั้นตอนดังแสดงในรูปที่ 3.4 โดยเริ่มต้นจากการสร้างชุดข้อมูลสำหรับฝึกที่มีเนื้อหาเกี่ยวกับภาควิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สจล. จากนั้นนำข้อมูลนี้ไปทำกระบวนการเตรียมข้อมูลเพื่อแปลงข้อมูลภาษาให้เป็นตัวเลขหรือโทเคน จากนั้นจึงจะนำชุดข้อมูลสำหรับฝึกนี้ไปเทรน WangchanBERTa Pre-trained Model ได้ หรือก็คือกระบวนการ Fine-tuning และสุดท้ายคือการประเมินประสิทธิภาพแบบจำลองโดยจะประเมินผ่านผลคะแนน Exact Match และ F1 – Score



รูปที่ 3.4 บล็อกไดอะแกรมการออกแบบแบบจำลองสำหรับระบบถามตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2.1 การจัดเตรียมชุดข้อมูลสำหรับฝึก ชุดข้อมูลประเมิน และชุดข้อมูลสำหรับทดสอบ

แบ่งสัดส่วนของข้อมูลออกเป็น ชุดข้อมูลสำหรับฝึก (Training Set) 80% , ชุดข้อมูลประเมิน (Validation Set) 10% , ชุดข้อมูลสำหรับทดสอบ (Test Set) 10% หรือ ชุดข้อมูลสำหรับฝึก : ชุดข้อมูลประเมิน : ชุดข้อมูลสำหรับทดสอบ = 80:10:10 โดยในปฏิญานิพนธ์นี้มีชุดข้อมูลทั้งหมด 375 ชุด ดังนั้นจึงสามารถแบ่งออกเป็น ชุดข้อมูลสำหรับฝึก 300 ชุดข้อมูล, ชุดข้อมูลประเมิน 38 ชุดข้อมูล และชุดข้อมูลสำหรับทดสอบ 37 ชุดข้อมูล

การสร้างชุดข้อมูลสำหรับฝึก ชุดข้อมูลประเมิน และชุดข้อมูลสำหรับทดสอบ จะมีรูปแบบเป็น List ของ Dictionary ดังแสดงในรูปที่ 3.5

```
[
  {
    "question_id": question id (int),
    "article_id": article id (int),
    "context": article texts (str),
    "question": [ question (str) ],
    "answers":
      {
        "text": [ answer text (str) ],
        "answer_start": [ answer beginning position (int) ],
        "answer_end": [ answer exclusive upper bound position (int) ],
      }
  },
]
```

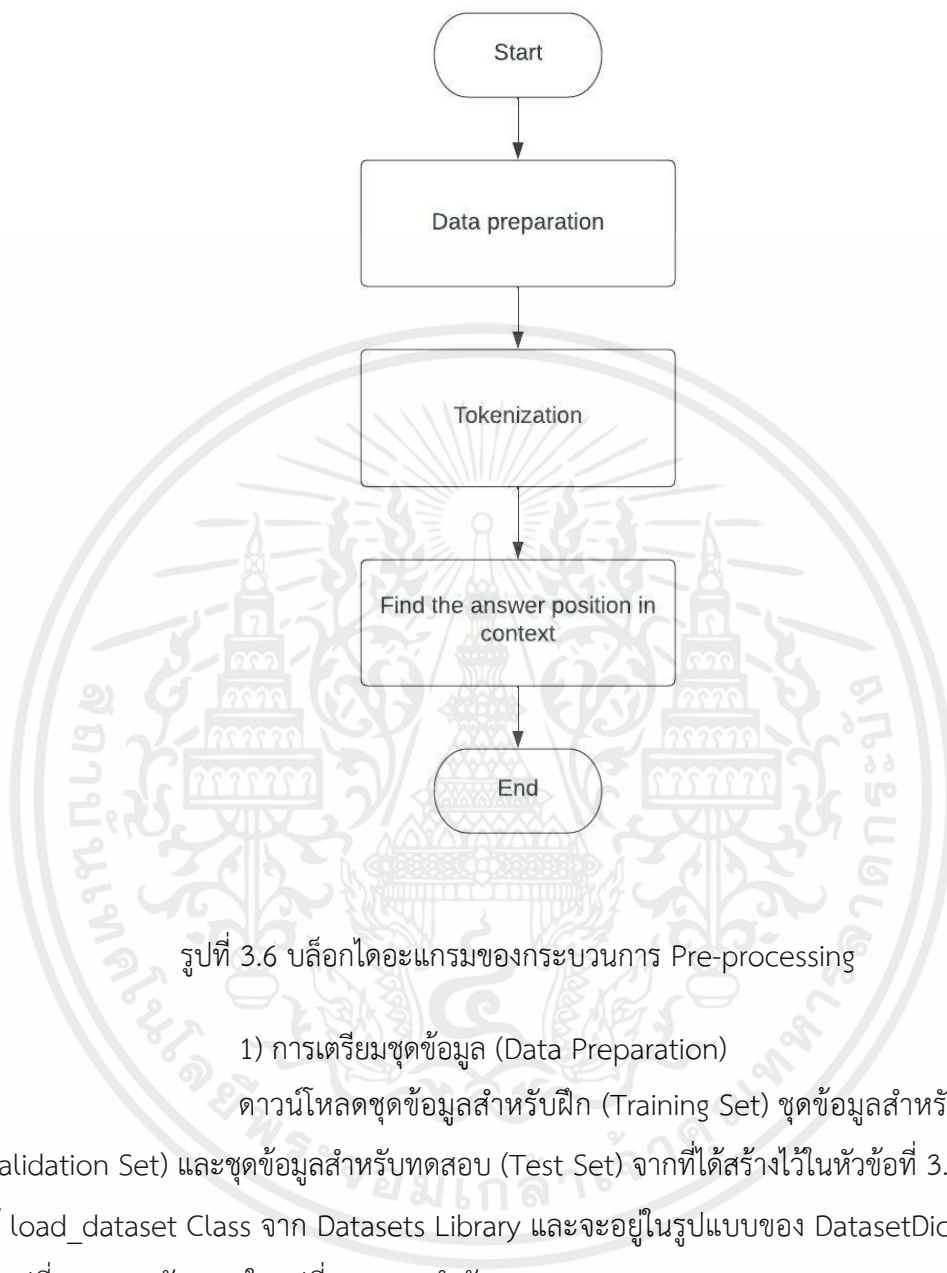
รูปที่ 3.5 รูปแบบของชุดข้อมูล (Dataset Format)

3.1.2.2 กระบวนการเตรียมข้อมูล (Pre-processing)

กระบวนการเตรียมข้อมูล (Pre-processing) จะมีขั้นตอนดังแสดงในรูปที่

3.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



1) การเตรียมชุดข้อมูล (Data Preparation)

ดาวน์โหลดชุดข้อมูลสำหรับฝึก (Training Set) ชุดข้อมูลสำหรับประเมิน (Validation Set) และชุดข้อมูลสำหรับทดสอบ (Test Set) จากที่ได้สร้างไว้ในหัวข้อที่ 3.1.2.1 โดยใช้ `load_dataset` Class จาก Datasets Library และจะอยู่ในรูปแบบของ `DatasetDict` ดังแสดงในรูปที่ 3.7 และดังแสดงในรูปที่ 3.8 ตามลำดับ

```
[26] 1 %cd /content/drive/MyDrive
      2
      3 from datasets import load_dataset
      4
      5 datasets = load_dataset(
      6     "json", data_files={
      7         "train": "train-dataV5.1.json",
      8         "validation": "valid-data-V1.json",
      9         "test": "test-data-V1.json"
     10     }
     11 )
```

รูปที่ 3.7 การดาวน์โหลดชุดข้อมูลสำหรับฝึก (Training Set) ชุดข้อมูลประเมิน (Validation Set) และชุดข้อมูลสำหรับทดสอบ (Test Set) ด้วย load_dataset Class

```
[27] 1 datasets
      DatasetDict({
      train: Dataset({
      features: ['question_id', 'article_id', 'context', 'question', 'answers'],
      num_rows: 302
      })
      validation: Dataset({
      features: ['question_id', 'article_id', 'context', 'question', 'answers'],
      num_rows: 55
      })
      test: Dataset({
      features: ['question_id', 'article_id', 'context', 'question', 'answers'],
      num_rows: 21
      })
      })
```

รูปที่ 3.8 ตัวแปร datasets ที่อยู่ในรูปแบบของ DatasetDict

2) Tokenization

ปัญญานิพนธ์นี้จะทำ Tokenization ด้วยการใช้ Class AutoTokenizer จาก Transformers Library ซึ่งเป็นการ Tokenize แบบเดียวกับ WangchanBERTa Pre-trained Model และจะเรียกใช้ from_pretrained() Method มาสร้าง Tokenizer Object ดังแสดงในรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ 3.9 ต่อมาจะเพิ่มข้อมูลคู่คำถาม (Question) และบริบท (Context) เข้าไปใน tokenizer Object ดังแสดงในรูปที่ 3.10

```
[6] 1 from transformers import AutoTokenizer
    2
    3 model_checkpoint = "airesearch/wangchanberta-base-att-spm-uncased"
    4 tokenizer = AutoTokenizer.from_pretrained(model_checkpoint)
```

รูปที่ 3.9 การสร้าง tokenizer Object

```
[16] 1 tokenized_datasets = tokenizer(
    2     datasets["question"],
    3     datasets["context"],
    4 )
```

รูปที่ 3.10 การเพิ่มข้อมูลคู่คำถามและบริบทใน tokenized_datasets

เนื่องจากในโครงการนี้จะต้อง Tokenize กับข้อความที่มีความยาวมาก จึงมีขั้นตอนของกระบวนการ Tokenization เพิ่มมาดังนี้

2.1 การตัดข้อความ (Truncation)

จากความจำ (Memory) ที่มีอย่างจำกัดของ Language Model จึงมีความจำเป็นอย่างมากที่จะต้องทำการตัดข้อความ (Truncation) ของชุดข้อมูลสำหรับฝึก (Training Set) ด้วยการเพิ่มพารามิเตอร์ truncation และ max_length ใน Tokenizer Object โดยจะกำหนดให้ truncation = "only_second" เนื่องจากจะ Truncate เพียงบริบท (Context) เท่านั้น และกำหนดให้ max_length = 416 เนื่องจาก WangchanBERTa Pre-trained Model รับความยาวได้ไม่เกิน 416 Token แสดงในรูปที่ 3.11

```
[ ] 1 tokenized_datasets = tokenizer(
2     datasets["question"],
3     datasets["context"],
4     truncation="only_second",
5     max_length=416,
6     return_overflowing_tokens=True,
7 )
```

รูปที่ 3.11 การแบ่งข้อมูล (Truncation) โดยกำหนด max_length = 416

2.2) การทับซ้อนกันของข้อความ (Overlapping)

ปัญหาที่ตามมาของการตัดข้อความ (Truncation) คือคำตอบที่อยู่ในบริบทอาจถูกแบ่งออกไปด้วยทำให้แบบจำลองต้องใช้เวลาอย่างมากในการหาคำตอบที่ถูกต้องจากบริบท [56] เพื่อแก้ไขปัญหานี้ จะอนุญาตให้บริบทที่ถูกแบ่งนั้น มีบางส่วนซ้ำกันหรือมีการซ้อนทับกันได้ (Overlapping) โดยการกำหนดพารามิเตอร์ return_overflowing_tokens = True และกำหนดความยาวที่จะให้ซ้อนทับด้วยพารามิเตอร์ stride เท่ากับความยาวที่ต้องการให้ซ้อนทับ หรือ stride = 128 ดังแสดงในรูปที่ 3.12

```
[ ] 1 tokenized_datasets = tokenizer(
2     datasets["question"],
3     datasets["context"],
4     truncation="only_second",
5     max_length=416,
6     return_overflowing_tokens=True,
7     stride=128,
8 )
```

รูปที่ 3.12 การทับซ้อนกันของข้อความ (Overlapping) โดยกำหนด stride = 128

2.3) การทำข้อความให้ยาวเท่ากัน (Padding)

คู่คำถามและบริบทที่นำมานั้นจะมีความยาวที่ไม่เท่ากันเสมอ ซึ่งจะไม่สามารถนำไปเทรนบน Pre-trained Model หรือการทำ Fine-tuning ได้ เนื่องจาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบจำลองต้องการอินพุตในรูปแบบ Uniform Shape ดังนั้นจึงต้องทำการ Padding คือการเติม Special Token <pad> ไปในคู่คำถามและบริบทที่มีความยาวน้อยกว่า max_length ให้ยาวเท่ากับ max_length หรือ 416 Token โดยใช้พารามิเตอร์ padding= “max_length” ดังแสดงในรูปที่ 3.13

```
[ ] 1 tokenized = tokenizer(
2     datasets["question"],
3     datasets["context"],
4     truncation="only_second",
5     max_length=416,
6     return_overflowing_tokens=True,
7     stride=128,
8     padding="max_length",
9     return_offsets_mapping=True,
10 )
```

รูปที่ 3.13 การทำข้อความให้ยาวเท่ากัน (Padding) โดยกำหนด padding = “max_length”

3) การหาตำแหน่งของช่วงคำตอบ (Answer Span)

คู่คำถามและบริบท (Question And Context Paired) เมื่อถูกป้อนเข้าสู่ AutoTokenizer แล้ว จำนวนของ Token ทั้งหมดที่มีความยาวมากกว่า max_length คู่คำถามและบริบทนี้จะถูกแบ่งออกเป็นหลาย Feature เนื่องมาจากการ Truncation เพื่อให้แต่ละ Feature มีความยาวไม่เกินค่า max_length ที่กำหนด จากคำอธิบายข้างต้น สามารถสรุปได้ว่า 1 คู่คำถามและบริบทสามารถมีค่าลักษณะได้มากกว่าหนึ่งค่าลักษณะสำหรับกรณีนี้ เนื่องจากการทำ Truncation บนบริบทและให้มีการซ้อนเหลื่อม (Overlapping) จะได้ว่าช่วงคำตอบ (Answer Span) จะอยู่ที่ค่าลักษณะใดค่าลักษณะหนึ่งเท่านั้น ดังนั้นจึงจำเป็นต้องระบุช่วงคำตอบว่าอยู่ที่ค่าลักษณะใด และอยู่มีตำแหน่งอยู่ที่ Token ไหน

จากหัวข้อที่ 3.1.2.2 เมื่อนำคู่คำถามและบริบทไปผ่าน AutoTokenizer Class เอาต์พุตที่ได้ จะประกอบไปด้วยข้อมูล ‘input_ids’, ‘attention_mask’, ‘offset_mapping’ และ ‘overflow_to_sample_mapping’ ซึ่งเอาต์พุตเหล่านี้จะถูกนำมาใช้ในการหาตำแหน่งของช่วงคำตอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยเอาต์พุตสุดท้ายที่ผู้จัดทำต้องการจากกระบวนการนี้ คือ 'input_ids', 'attention_mask', 'start_position' และ 'end_position'

อันดับแรกผู้จัดทำจะทำการตรวจสอบว่าคำถามนั้น ๆ ได้มีการกำหนดคำตอบไว้หรือไม่ โดยแบ่งออกเป็นกรณีดังนี้

1) สำหรับคำถามที่ไม่ได้มีการกำหนดคำตอบ ผู้จัดทำจะกำหนดให้ 'start_position' และ 'end_position' มีค่าเท่ากับตำแหน่งของ Token <s> หรือมีค่าเท่ากับ 0

2) สำหรับคำถามที่มีการกำหนดคำตอบ ผู้จัดทำจะทำการหาตำแหน่งของ Character ที่เป็นจุดเริ่มต้นของบริบท มาเก็บไว้ในตัวแปร start_char และหา Character สุดท้ายของบริบทมาเก็บในตัวแปร end_char จากชุดข้อมูลสำหรับฝึก (Training Set) ต่อมาจะหาตำแหน่งของ Token ที่เป็นจุดเริ่มต้นของบริบทมาเก็บในตัวแปร token_start_index และหาตำแหน่งของ Token สุดท้ายของบริบทเก็บในตัวแปร token_end_index ด้วยฟังก์ชัน _get_context_span ดังแสดงในรูปที่ 3.14 จากนั้นเอาต์พุตของฟังก์ชันจะถูกนำไปใช้ในการหา 'start_position' และ 'end_position' ด้วยฟังก์ชัน _get_answer_span ดังแสดงในรูปที่ 3.15

```
def _get_context_span(input_ids,
                    sequence_ids,
                    answers,
                    start_col,
                    text_col='text',
                    pad_on_right=True):
    # Start/end character index of the answer in the text.
    start_char = answers[start_col][0]
    end_char = start_char + len(answers[text_col][0]) + 1

    # Start token index of the current span in the text.
    token_start_index = 0
    while sequence_ids[token_start_index] != (1 if pad_on_right else 0):
        token_start_index += 1

    # End token index of the current span in the text.
    token_end_index = len(input_ids) - 1
    while sequence_ids[token_end_index] != (1 if pad_on_right else 0):
        token_end_index -= 1

    return token_start_index, token_end_index, start_char, end_char
```

รูปที่ 3.14 การหา start_char, end_char, token_start_index และ token_end_index

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def _get_answer_span(tokenized_examples,
                    offsets,
                    start_char,
                    end_char,
                    token_start_index,
                    token_end_index,):

    while token_start_index < len(offsets) and offsets[token_start_index][0] <= start_char:
        token_start_index += 1
    start_position = token_start_index - 1
    while offsets[token_end_index][1] >= end_char:
        token_end_index -= 1
    end_position = token_end_index + 1

    return start_position, end_position

```

รูปที่ 3.15 การหา 'start_position' และ 'end_position'

3.1.2.3 Hyperparameter Optimization

สำหรับกระบวนการทำ Hyperparameter Optimization หรือ Hyperparameter Tuning นี้ จะเป็นการหาชุด (Set) ของ Hyperparameters ที่เหมาะสมสำหรับแบบจำลอง ที่จะส่งผลให้แบบจำลองมีประสิทธิภาพ มีความแม่นยำ (Accuracy) ที่สูงขึ้น หรือต้องการลดค่า Loss ให้มีค่าต่ำที่สุด โดยผู้จัดทำได้เลือกใช้ Optuna [57] ซึ่งเป็น Automated Hyperparameter Tuning

สำหรับ Object Parameter ต่าง ๆ ที่ใช้ในการ Optimization ด้วย Optuna ผู้จัดทำได้ทำการกำหนดให้มีค่าดังรูปที่ 3.16 โดย LR_MIN และ LR_CEIL คือ Minimum Learning Rate และ Maximum (Ceiling) Learning Rate ตามลำดับ WD_MIN และ WD_CEIL คือ Minimum Weight Decay และ Maximum Weight Decay ตามลำดับ MIN_EPOCHS และ MAX_EPOCHS คือ Minimum Epoch และ Maximum Epoch ตามลำดับ PER_DEVICE_EVAL_BATCH คือ ขนาดของ Batch Size ต่อ GPU/CPU core สำหรับการ Evaluation PER_DEVICE_TRAIN_BATCH คือ ขนาดของ Batch Size ต่อ GPU/CPU core สำหรับการเทรน NUM_TRIALS คือจำนวนของการทดลอง (Trial) SAVE_DIR คือชื่อของ Folder ที่ต้องการบันทึก NAME_OF_MODEL คือชื่อของแบบจำลองที่จะใช้ในการ Optimization และสุดท้าย MAX_LENGTH คือความยาวสูงสุดของลำดับประโยค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

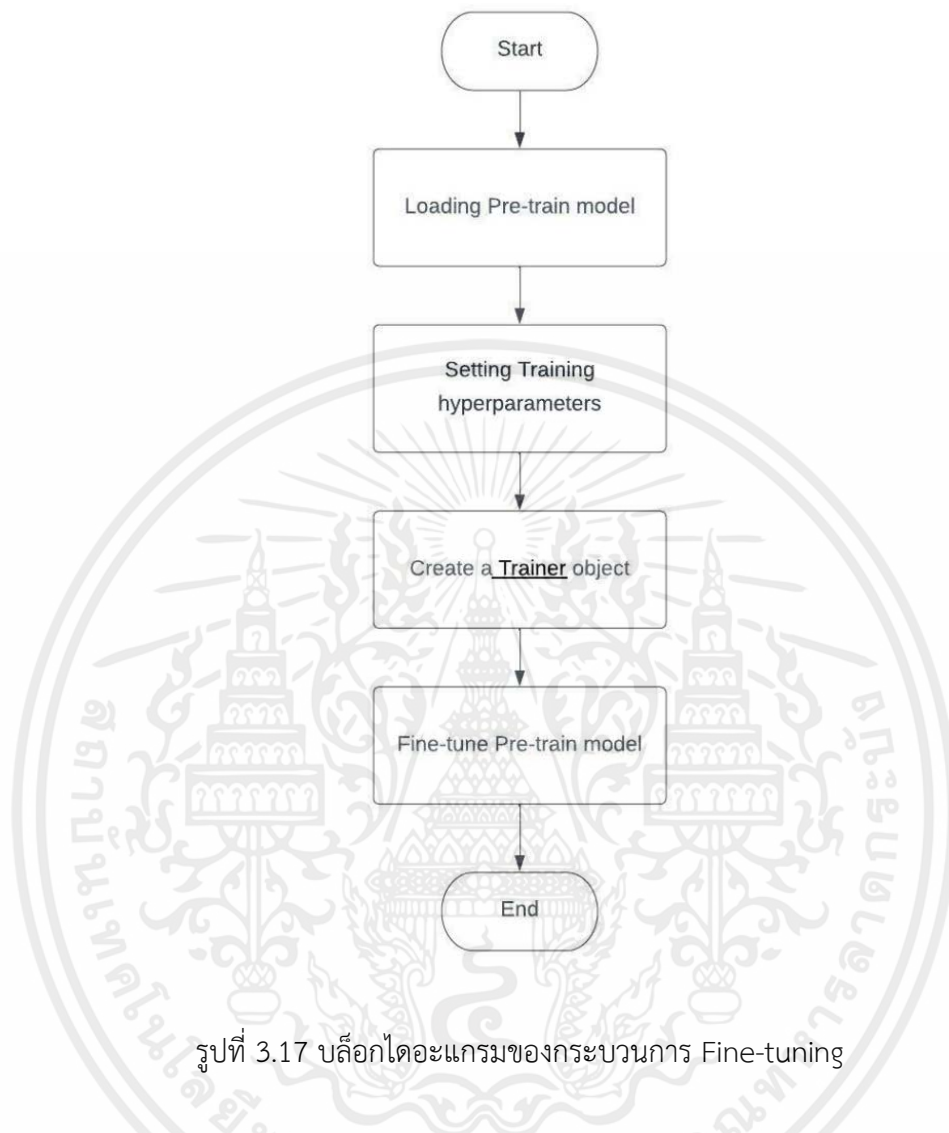
#Object Parameters
LR_MIN = 4e-5
LR_CEIL = 0.01
WD_MIN = 4e-5
WD_CEIL = 0.01
MIN_EPOCHS = 2
MAX_EPOCHS = 5
PER_DEVICE_EVAL_BATCH = 8
PER_DEVICE_TRAIN_BATCH = 16
NUM_TRIALS = 10
SAVE_DIR = 'opt-test'
NAME_OF_MODEL = 'finetune_telecom_thaiqa'
MAX_LENGTH = 416

```

รูปที่ 3.16 Object Parameter ต่าง ๆ ที่ใช้ในการ Optimization ด้วย Optuna

3.1.2.4 กระบวนการ Fine-tuning

กระบวนการ Fine-tuning หรือการเทรน Pre-trained Model จะมีขั้นตอนดังแสดงในรูปที่ 3.17



รูปที่ 3.17 บล็อกไดอะแกรมของกระบวนการ Fine-tuning

1) ดาวน์โหลด Pre-trained Model ที่จะนำมาเทรน

ในปริณญาณิพนธ์นี้จะเลือกใช้ `airesearch/wangchanberta-base-att-spm-uncased` เนื่องจากเป็นแบบจำลองแบบภาษาเดียว (Monolingual Language Model) บนชุดข้อมูลภาษาไทยขนาดกว่า 78.5 GB [58] ซึ่งใหญ่ที่สุดของประเทศไทยในปัจจุบันนี้ โดยสามารถดาวน์โหลด Pre-trained Model นี้ได้จาก `AutoModelForQuestionAnswering Class` และเรียกใช้ผ่าน `from_pretrained()` Method ดังแสดงในรูปที่ 3.18

```
[ ] 1 from transformers import AutoModelForQuestionAnswering
    2
    3 model_checkpoint = "airesearch/wangchanberta-base-att-spm-uncased"
    4 model = AutoModelForQuestionAnswering.from_pretrained(model_checkpoint)
```

รูปที่ 3.18 การดาวน์โหลด Pre-trained Model ด้วย AutoModelForQuestionAnswering Class

2) ไฮเปอร์พารามิเตอร์สำหรับการเทรน (Training Hyperparameter)

เมื่อได้พารามิเตอร์ที่ดีที่สุดจากการทำกระบวนการ Hyperparameter Optimization จากหัวข้อที่ 3.1.2.3 แล้ว พารามิเตอร์ดังกล่าวจะถูกนำมาใช้ในการเทรนบน Pre-trained Model เพื่อให้ได้แบบจำลองสุดท้ายมา (Final Model) โดย Hyperparameter ต่าง ๆ จะถูกกำหนดผ่าน TrainingArguments Class แสดงดังรูปที่ 3.17

```
[ ] 1 from transformer import TrainingArguments
    2
    3 args = TrainingArguments(
    4     f"finetune_telecom_thaiqa",
    5     evaluation_strategy = "epoch",
    6     learning_rate=8e-5,
    7     per_device_train_batch_size=16,
    8     per_device_eval_batch_size=8,
    9     num_train_epochs=3,
    10    weight_decay=0.00013,
    11    save_steps=2000,
    12 )
```

รูปที่ 3.19 การกำหนด Hyperparameter ผ่าน TrainingArguments

3) การสร้าง Trainer Object และเทรนแบบจำลองด้วย Trainer

ในปริณญาณิพนธ์นี้จะใช้ Trainer ในการเทรนแบบจำลองโดย Trainer เป็น Class หนึ่งใน Transformers Library ถูกปรับแต่งมาจาก PyTorch เพื่อให้ง่ายต่อการเทรนแบบจำลอง โดยไม่ต้องเขียนลูปการฝึกสอน (Training Loop) เอง [59] แต่ก่อนที่จะไปเทรน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pre-train Model จะต้องนำชุดข้อมูลสำหรับฝึก (Training Set), ชุดข้อมูลสำหรับประเมิน (Validation Set), TrainingArguments และ Pre-train Model ที่เลือกไว้มาสร้างเป็น trainer Object ก่อน

นอกจากนี้ต้องเพิ่มพารามิเตอร์ tokenizer เพื่อบันทึก Tokenizer ที่ใช้ในขั้นตอนการเตรียมข้อมูล (Pre-processing) ดังหัวข้อที่ 3.1.4 ไว้ในแบบจำลองนี้ด้วย เพื่อให้ง่ายต่อการเรียกใช้งานในครั้งต่อไป [60] และเพิ่ม data_collator โดยการเรียกใช้ Class DefaultDataCollator เพื่อเปลี่ยนรูปแบบของชุดข้อมูลให้อยู่ในรูปแบบที่สามารถเรนได้ดังแสดงในรูปที่ 3.20 แล้วจึงจะนำไปเรน Pre-train Model ที่เลือกไว้ด้วยการใช้คำสั่ง train() ดังแสดงในรูปที่ 3.21

```
[ ] 1 from transformers import Trainer, DefaultDataCollator
2
3 data_collator = DefaultDataCollator
4
5 trainer = Trainer(
6     model=model,
7     args=args,
8     train_dataset=tokenized_datasets["train"],
9     eval_dataset=tokenized_datasets["validation"],
10    data_collator=data_collator,
11    tokenizer=tokenizer,
12 )
```

รูปที่ 3.20 การสร้าง Trainer Object

```
[ ] 1 trainer.train()
```

รูปที่ 3.21 การเรน Pre-train Model ด้วยคำสั่ง train()

3.1.2.5 กระประเมินประสิทธิภาพแบบจำลอง (Evaluation)

สำหรับการประเมินประสิทธิภาพแบบจำลอง (Evaluation) หรือประเมินประสิทธิภาพการคาดการณ์ (Prediction) คำตอบของแบบจำลอง ชุดข้อมูลสำหรับทดสอบ (Test Set) จะถูกนำมาใช้ และผู้จัดทำได้เลือกใช้ฟังก์ชัน question_answering_metrics จาก thai2transformers [25] ในการประเมินประสิทธิภาพแบบจำลองในครั้งนี้ ดังแสดงในรูปที่ 3.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#using thai2transformers
from thai2transformers.metrics import question_answering_metrics
metrics, formatted_predictions, references = question_answering_metrics(datasets['test'], trainer)
```

รูปที่ 3.22 การประเมินประสิทธิภาพแบบจำลองด้วยฟังก์ชัน question_answering_metrics จาก thai2transformers

3.1.3 การสร้างและออกแบบ Knowledge Base

สำหรับกระบวนการนี้ ผู้จัดทำจะทำการสร้าง Knowledge Base ขึ้น ซึ่งเป็นฐานข้อมูลหรือ Database รูปแบบ RDBMS (Relational Database Management System) โดยแบ่งออกเป็น 2 ตาราง หนึ่งตารางสำหรับเก็บข้อมูลบริบท และอีกหนึ่งตารางสำหรับเก็บคำถามจากผู้ใช้ที่ระบบไม่สามารถตอบได้ ฐานข้อมูลดังกล่าวจะถูก Host ไว้บน Google Cloud Sever ผ่านบริการ Cloud SQL โดยมีขั้นตอนและกระบวนการ ดังนี้

3.1.3.1 การสร้าง Database Instance

สำหรับ Database Instance หรือ VM Instance เป็นการสร้างเครื่องเซิร์ฟเวอร์ (Virtual Machine) โดยสามารถกำหนดได้ตั้งแต่ฮาร์ดแวร์ของเครื่อง ไปจนถึง OS ที่ต้องการ เปรียบเสมือนการสร้างเครื่องเซิร์ฟเวอร์เป็นของตัวเอง โดย Google มี Database Engine ให้เลือกได้แก่ MySQL, PostgreSQL และ SQL Server ซึ่งผู้จัดทำได้เลือกใช้ MySQL และ Database Version ที่เลือกใช้คือ MySQL 8.0 สำหรับ Machine Type เป็นแบบ Shared Core 1 vCPU, 1.7 GB, Storage Capacity 10 GB และการกำหนดการเชื่อมต่อ (Connections) จะเป็นการเชื่อมต่อผ่าน Public IP ซึ่งต้องกำหนดที่อยู่ IP ภายนอก (External IP) ที่เข้าถึงอินเทอร์เน็ตได้ ต้องใช้เครือข่ายที่ได้รับอนุญาต (Authorized Networks) หรือ Cloud SQL Proxy เพื่อเชื่อมต่อกับกรณีตัวอย่าง (Instance) ดังรูปที่ 3.23 แสดงการตั้งค่า Database Instance โดยรวม และเมื่อทำการสร้างกรณีตัวอย่างเสร็จแล้ว จะได้ Public IP Address ของ Database มา แสดงดังรูปที่ 3.24 ซึ่ง Public IP Address ดังกล่าวจะถูกนำไปใช้เพื่อให้ Database Visualization Tool สามารถเชื่อมต่อ/ติดต่อกับ Database ได้

Region	asia-southeast1 (Singapore)
DB Version	MySQL 8.0.26
vCPUs	1 vCPU
Memory	1.7 GB
Storage	10 GB
Network throughput (MB/s) ?	125 of 125
Disk throughput (MB/s) ?	Read: 4.8 of 125.0 Write: 4.8 of 37.9
IOPS ?	Read: 300 of 15,000 Write: 300 of 4,500
Connections	Private IP Public IP
Backup	Automated
Availability	Single zone
Point-in-time recovery	Enabled

รูปที่ 3.23 การตั้งค่า Database Instance โดยรวม

<input type="checkbox"/>	Instance ID ? ↑	Type	Public IP address	Private IP address	Instance connection name
<input checked="" type="checkbox"/>	telecom-qa	MySQL 8.0	35.186.147.213	10.84.224.3	telecom-qa-app:asia... ▼

รูปที่ 3.24 Public IP Address ของ Database ที่ได้

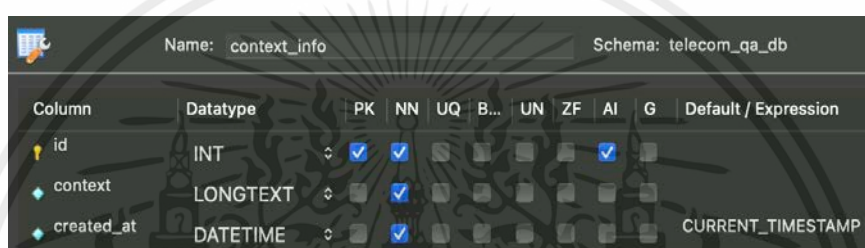
3.1.3.2 การสร้างฐานข้อมูล (Database)

สำหรับการสร้าง ติดต่อ และจัดการฐานข้อมูล สามารถสร้างและจัดการผ่าน Google Cloud Console หรือผ่านการใช้ Database Visualization Tool ก็ได้ โดยผู้จัดทำใช้ Database Visualization Tool ซึ่งในที่นี้คือ MySQL Workbench เพื่อเข้าถึงและแก้ไขฐานข้อมูล

1) การสร้างและออกแบบตารางสำหรับเก็บข้อมูลบริษัท

ตารางสำหรับเก็บข้อมูลบริษัทถูกสร้างขึ้นเพื่อเก็บข้อมูลบริษัทหรือคลังความรู้ทั้งหมดไว้ ข้อมูลบริษัทต่าง ๆ ที่ถูกเก็บไว้ในตารางนี้ จะถูก Retriever ดึงไปใช้ เพื่อหาบริษัทที่สัมพันธ์กับคำถามมากที่สุด โดยข้อมูล (Field/Column) ที่ถูกเก็บไว้ในตาราง ได้แก่ ID ของแต่ละ

บริบท (id), บริบทต่าง ๆ (context) และวัน/เวลาที่ทำการเพิ่มข้อมูล (created_at) ซึ่งผู้จัดทำได้กำหนดเงื่อนไข (Constraint) ในการอนุญาตให้เก็บเฉพาะข้อมูลที่เหมาะสมลงในฐานข้อมูลเพื่อให้อ่านข้อมูลจากฐานข้อมูลเป็นไปอย่างถูกต้องไว้ ดังรูปที่ 3.25 โดย id ถูกกำหนดให้เป็น PK หรือ Primary Key, Not NULL และ Auto Increment, context ถูกกำหนดเงื่อนไขบังคับให้เป็น Not NULL และสุดท้าย created_at ถูกกำหนดเงื่อนไขให้เป็น Not NULL และถูกตั้งค่า Default/Expression เป็น CURRENT_TIMESTAMP ซึ่งเป็นฟังก์ชันที่จะส่งคืนวันและเวลาปัจจุบันใน Format ของ ‘ YYYY-MM-DD hh:mm:ss.mmm ’



Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
context	LONGTEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
created_at	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP

รูปที่ 3.25 การกำหนด Constraint ให้กับแต่ละ Column ในตารางสำหรับเก็บข้อมูลบริบท

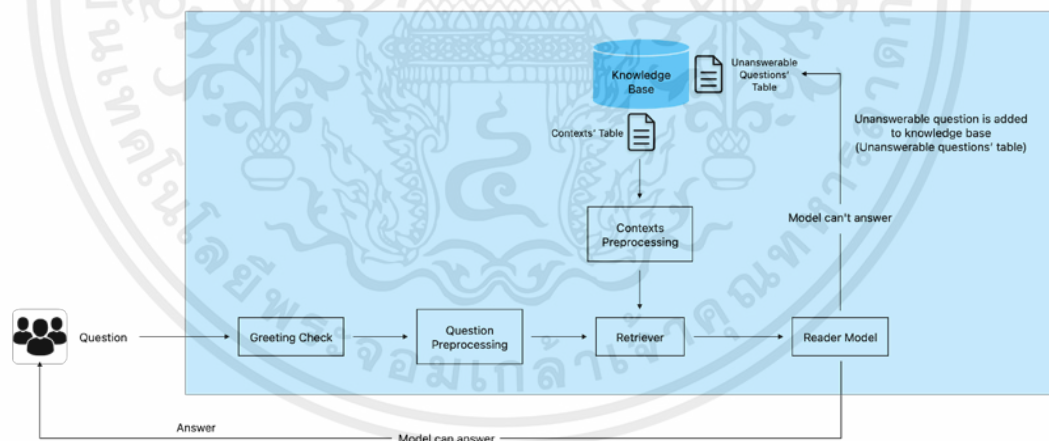
2) การสร้างและออกแบบตารางสำหรับคำถามที่ระบบไม่สามารถตอบได้ ตารางสำหรับเก็บข้อมูลสำหรับคำถามที่ระบบไม่สามารถตอบได้ ถูกสร้างขึ้นเพื่อเก็บข้อมูลคำถามจากผู้ใช้งานที่ซึ่งระบบไม่สามารถตอบได้ เพื่อนำคำถามเหล่านี้ไปตรวจสอบว่าคำถามหรือเรื่องใดที่ผู้ใช้ต้องการทราบ และจะถูกนำไปปรับปรุงประสิทธิภาพของระบบเพื่อให้สามารถตอบคำถามได้ดีขึ้น โดยข้อมูล (Field/Column) ที่ถูกเก็บไว้ในตาราง ได้แก่ ID ของแต่ละคำถาม (id) , คำถาม (question) และวัน/เวลาที่ทำการเพิ่มข้อมูล (inserted_at) ซึ่งผู้จัดทำได้กำหนดเงื่อนไขไว้ดังแสดงในรูปที่ 3.26 โดย id ถูกกำหนดเงื่อนไขเป็น PK หรือ Primary Key, Not NULL และ Auto Increment, question ถูกกำหนดเงื่อนไขให้เป็น Not NULL และสุดท้าย inserted_at ถูกกำหนดเงื่อนไขบังคับให้เป็น Not NULL และถูกตั้งค่า Default/Expression เป็น CURRENT_TIMESTAMP

Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
question	LONGTEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
inserted_at	DATETIME	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	CURRENT_TIMESTAMP

รูปที่ 3.26 การกำหนดเงื่อนไขบังคับให้กับแต่ละ Column ในตารางสำหรับคำถามที่ระบบไม่สามารถตอบได้

3.1.4 การสร้างและออกแบบระบบของ Recommendation System API

ในการออกแบบ Recommendation System มีวัตถุประสงค์เพื่อสร้างเป็น API แล้วเปิดให้บริการให้กับผู้พัฒนาเว็บแอปพลิเคชัน หรือหน่วยงานที่ต้องการใช้ระบบถาม-ตอบด้วยข้อมูลของตนเอง แต่ในปฏิญญาฉบับนี้จะนำระบบมาใช้ในการถาม-ตอบในเรื่องของหลักสูตรวิศวกรรมโทรคมนาคม สาขาวิศวกรรมโทรคมนาคมและโครงข่าย ซึ่งจะมีส่วนการทำงานต่าง ๆ ดังแสดงในรูปที่ 3.27



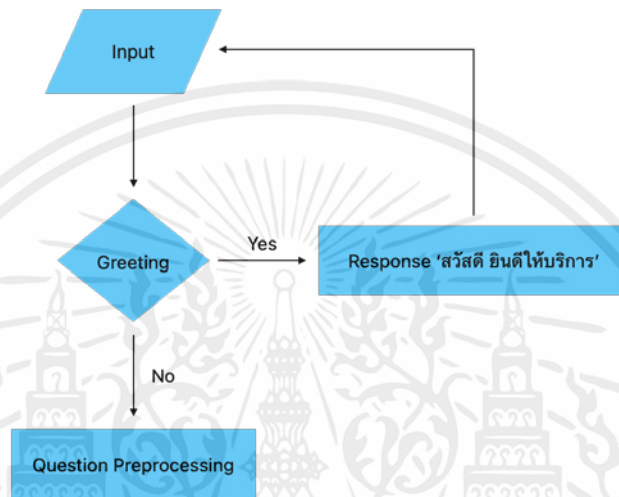
รูปที่ 3.27 บล็อกไดอะแกรมแสดงการทำงานของ Recommendation System API

3.1.4.1 ระบบตรวจสอบการทักทาย (Greeting Check)

ระบบตรวจสอบการทักทาย มีวัตถุประสงค์เพื่อใช้ในการตรวจสอบข้อความที่ผู้ใช้ป้อนเข้าสู่ระบบว่าข้อความดังกล่าวเป็นประโยคคำถาม หรือทักทาย หากเป็นการทักทาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบจะทำการตรวจจับ (Detect) คำต่อไปนี้จากประโยคที่ผู้ใช้ป้อนเข้าสู่ระบบ 'สวัสดีครับ', 'สวัสดีค่ะ', 'สวัสดี', 'หวัดดีครับ', 'หวัดดีค่ะ', 'หวัดดี', 'Hi', 'hi', 'hello', 'Hello', 'ฮัลโหล', 'สวัสดีจ้า', 'ดีอะ' และ 'ดีจ้า' โดยเมื่อระบบตรวจพบคำดังกล่าว ระบบจะทำการโต้ตอบกับผู้ใช้ด้วยความ 'สวัสดี ยินดีให้บริการ' โดยแสดงการทำงานดังรูปที่ 3.28



รูปที่ 3.28 การทำงานของระบบตรวจสอบการทักทาย

3.1.4.2 ระบบเตรียมข้อมูลคำถาม (Question Pre-processing)

เนื่องจากคอมพิวเตอร์ประมวลผลและเข้าใจได้จากข้อมูลที่เป็นตัวเลข ไม่สามารถเข้าใจตัวอักษรหรือภาษามนุษย์ได้ การแปลงภาษาให้เป็นตัวเลขจึงเป็นกลไกแรกในการประมวลผลภาษา แต่ข้อความที่ได้นั้น บางทีอาจจะนำมาจากอินเทอร์เน็ต ซึ่งจะมีข้อความที่ไม่จำเป็น เช่น แท็ก HTML, เครื่องหมายวรรคตอน รวมไปถึงอักษร หรือคำที่ไม่มีความหมาย ดังนั้นก่อนจะนำคำถามจากผู้ใช้งานไปทำกระบวนการต่อไป จึงต้องทำกระบวนการ Preprocessing ก่อนแสดงการทำงานภายในระบบเตรียมข้อมูลคำถาม แสดงดังรูปที่ 3.29 โดย Tokenization หรือการตัดคำ เป็นการแบ่งข้อความออกเป็นคำย่อย ๆ ซึ่งจะมีการแปลงจากแต่ละคำให้เป็นตัวเลข ผ่านการใช้ฟังก์ชัน input_ids และเนื่องจากเอาต์พุตที่ได้จากฟังก์ชันนี้ มีการนับช่องว่าง และ Special Token ด้วย จึงจำเป็นต้องทำการกำจัดช่องว่างและ Special Token ออก แล้วจึงแปลงจากตัวเลขให้กลับมามีอยู่ในรูปของคำ และเมื่อทำการแบ่งข้อความออกเป็นคำย่อย ๆ เป็นที่เรียบร้อยแล้ว จะเข้าสู่กระบวนการสุดท้ายของระบบเตรียมข้อมูลคำถาม คือกระบวนการกำจัดคำฟุ่มเฟือย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Remove Stop Word) ซึ่งเป็นการลบคำที่ไม่สื่อความหมาย หรือยังสื่อความหมายเดิมออกจากประโยค โดยจะทำการลบคำหยุดด้วยฟังก์ชัน `thai_stopwords` จากมอดูล `pythainlp.corpus` และอีกหนึ่งวัตถุประสงค์ของการกำจัดคำฟุ่มเฟือย เนื่องจากคำถามจากผู้ใช้ที่ผ่านการเตรียม หรือ Preprocess เป็นที่เรียบร้อยแล้ว จะถูกนำไปใช้ใน Retriever ร่วมกับข้อมูลบริบทที่ผ่านการเตรียมข้อมูลแล้ว เพื่อหาบริบทที่เกี่ยวข้องกับคำถามมากที่สุด ผ่านการใช้อัลกอริทึม Okapi BM25 ซึ่งอัลกอริทึมดังกล่าวจะตรวจสอบความถี่คำที่ซ้ำกันของคำถามและบริบท ดังนั้นจึงจำเป็นต้องลบคำฟุ่มเฟือยออก เพื่อไม่ให้มีคำที่ไม่จำเป็นถูกนำไปหาความถี่คำ เนื่องจากอาจทำให้อัลกอริทึม Okapi BM25 หาบริบทที่เกี่ยวข้องไม่เจอ โดยสำหรับกระบวนการต่าง ๆ ภายใน Retriever จะอธิบายเพิ่มเติมในหัวข้อที่ 3.1.4.3



รูปที่ 3.29 การทำงานภายในระบบเตรียมข้อมูลคำถาม

3.1.4.3 ระบบเตรียมข้อมูลบริบท (Contexts Preprocessing)

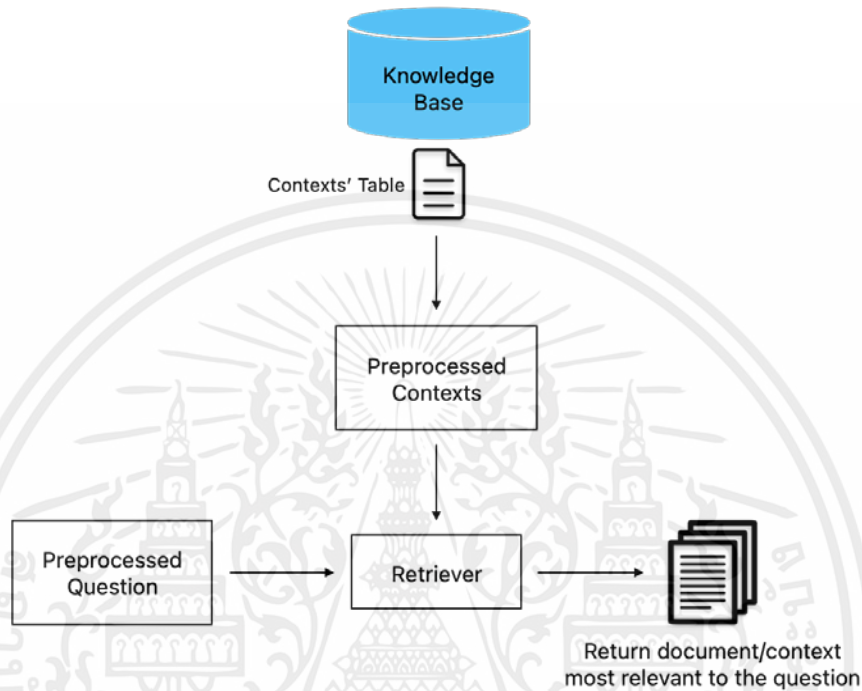
ข้อมูลบริบทจะถูกดึงมาจากตารางสำหรับเก็บข้อมูลบริบท ใน Knowledge Base และจะถูกนำมาผ่านกระบวนการเตรียมข้อมูลด้วยระบบเตรียมข้อมูลบริบท ซึ่งสำหรับระบบเตรียมข้อมูลบริบทนั้น กระบวนการภายในจะมีเพียงแค่การทำ Tokenization เท่านั้น เนื่องจากใช้ในการ Initialize อัลกอริทึม Okapi BM25 เพียงเท่านั้น ซึ่งต่างจากระบบเตรียมข้อมูลคำถาม

3.1.4.4 ระบบค้นคืนข้อมูล (Retriever)

ระบบสืบค้นคืนข้อมูล หรือ Retriever เป็นระบบที่ทำหน้าที่ค้นหาบริบทที่เกี่ยวข้อง จากคำถามของผู้ใช้งาน โดยจะใช้อัลกอริทึม Okapi BM25 ซึ่งจะมีวิธีการคือเปรียบเทียบความถี่ของคำจากคำถามกับบริบททั้งหมดที่ถูกเก็บไว้ใน Knowledge Base แล้วส่งคืนบริบทที่เกี่ยวข้องกับคำถามมากที่สุดกลับมา

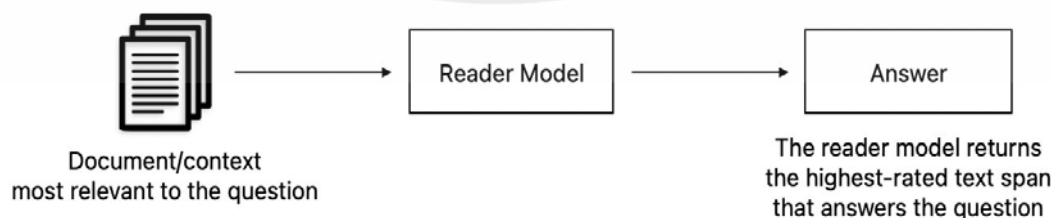
โดยระบบจะทำการดึงข้อมูลบริบททั้งหมดที่ถูกเก็บไว้ใน Knowledge Base และมาทำการ Tokenization ก่อน แล้วจึงทำการสร้างกรณีตัวอย่างจาก `rank_bm25 Class` ก่อน จากนั้นจะนำคำถามของผู้ใช้งานที่ผ่านกระบวนการ Preprocessing แล้ว มาทำการค้นหา

บริบทที่เกี่ยวข้องมากที่สุด แล้วส่งคืนกลับมาบล็อกไดอะแกรมแสดงการทำงานภายในระบบค้นคืนข้อมูล แสดงดังรูปที่ 3.30



รูปที่ 3.30 การทำงานภายในระบบค้นคืนข้อมูล (Retriever)

3.1.4.5 การใช้งาน Reader Model ใน Recommendation System API
 เมื่อได้บริบทที่เกี่ยวข้องกับคำถามจากผู้ใช้มากที่สุดจากระบบค้นคืนข้อมูลมาแล้ว จะนำ Reader Model ที่ได้สร้างขึ้นในหัวข้อที่ 3.1.2 มาทำการประมวลผลคำถามและบริบทดังกล่าวเพื่อคาดเดา (Predict) ช่วงของคำตอบ ดังแสดงในรูปที่ 3.31



รูปที่ 3.31 การทำงานของ Reader Model

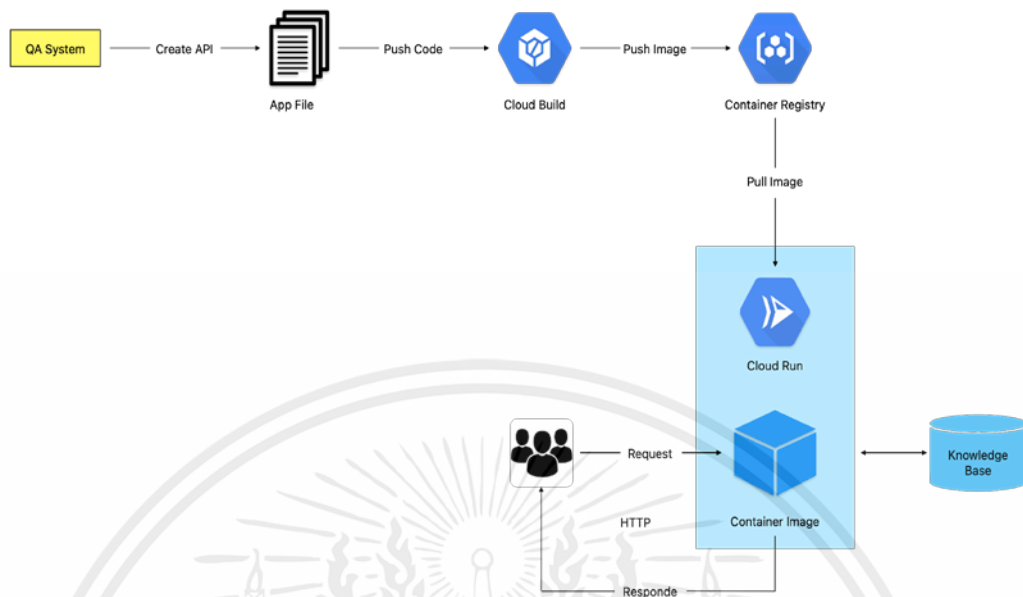
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.5 การ Deploy Recommendation System บน Google Cloud

เมื่อสร้างระบบขึ้นมาได้ตามที่ต้องการ จากนั้นจะนำไปสู่ขั้นตอนของการนำไปใช้งานจริง หรือที่เรียกว่า Deployment สำหรับกระบวนการนี้จะทำการ Deploy ระบบบน Production โดยจะนำระบบไป Deploy บน Cloud Server ซึ่งผู้จัดทำได้เลือกใช้บริการจาก Google Cloud เพื่อให้ผู้ใช้สามารถเรียกใช้งานผ่านโพรโทคอล HTTP โดยมีเป้าหมายให้ผู้ใช้สามารถเรียกใช้งานระบบผ่าน API ได้

โดยผู้จัดทำได้เลือกใช้ FastAPI ในการสร้าง API เนื่องจากมีความรวดเร็วในการทำงาน ประสิทธิภาพเทียบเท่ากับ Node.js และ Go ง่ายต่อการใช้ และ Uvicorn จะทำหน้าที่เป็นเซิร์ฟเวอร์ในการรัน Python Application แบบวิธีการ Asynchronous โดยมี /predict เป็น API Endpoint (Route) ซึ่งผู้จัดทำได้กำหนดไว้ (<https://hostname/predict>) และมีการรับข้อมูลเป็น JSON Format จาก HTTP POST Method และส่งผลการ Predict เป็น JSON Format เช่นเดียวกัน

หลังจากนั้นจะเป็นกระบวนการของการนำระบบขึ้น Cloud Server ซึ่ง Docker จะถูกนำมาใช้ในการทำหน้าที่จำลองสภาพแวดล้อมเฉพาะให้กับซอฟต์แวร์หรือตัวระบบ เพื่อให้ซอฟต์แวร์สามารถทำงานได้แม้อยู่ในสภาพแวดล้อมอื่น โดยผู้จัดทำใช้บริการ Cloud Build ในการสร้าง Docker Image ผ่าน Dockerfile และ Image จะถูก Push ไปที่ Google Container Registry (GCR) ซึ่งเป็นการลงทะเบียนแบบส่วนตัวสำหรับเก็บ Image ดังเช่น Docker Image เพื่อนำไปสร้างกรณีตัวอย่างโดยสามารถใช้ Cloud Build ในการ Deploy Container Image จาก Container Registry ไปที่ Cloud Run ได้เลย ผ่านการใช้ Container Image URL โดย Cloud Run จะนำ Container Image ที่เปิดให้ใช้งานผ่านทาง HTTP ไปรันให้ [61] ซึ่งผู้ใช้สามารถเข้าถึงหรือใช้งานแอปพลิเคชันแบบออนไลน์ผ่าน HTTP จาก API ได้เลยทันที โดยบล็อกไดอะแกรมแสดงกระบวนการ Deploy และการทำงานโดยรวม แสดงดังรูปที่ 3.32



รูปที่ 3.32 กระบวนการ Deploy และการทำงานโดยรวม

3.1.6 การออกแบบกราฟิกเคลื่อนไหว (Motion Graphic)

กราฟิกเคลื่อนไหวเป็นรูปแบบงานกราฟิกที่สร้างจากภาพวาด 2 มิติ โดยมีระบบการทำงานในรูปแบบ Animation ที่สร้างภาพเคลื่อนไหวให้ปรากฏบนจอภาพ ซึ่งอาจจะประกอบด้วย การใส่เสียงและสื่อผสมต่าง ๆ มีวิธีการออกแบบ ดังนี้

- 1) ออกแบบตัวละครจากเอกลักษณ์ของคณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมโทรคมนาคมและโครงข่าย นั่นก็คือเสาสื่อสารหรือที่เรียกกันว่าเสาโทรคมนาคม มาเป็นจุดเด่นของตัวละคร แสดงดังรูปที่ 3.33



รูปที่ 3.33 เสาโทรคมนาคม ณ ตึกคณะวิศวกรรมศาสตร์ สาขาวิชาวิศวกรรมโทรคมนาคมและ
โครงข่าย

2) ออกแบบตัวละครโดยการคำนึงถึงสี รูปร่าง และลักษณะเฉพาะของตัวละคร เพื่อให้ตัวละครนั้นมีลักษณะที่จดจำได้ง่าย โดยใช้โปรแกรม Clip-Studio Paint สำหรับวาดตัวละคร โดยตัวละครนี้มีชื่อว่า “น้องเทเลคอมจิ้ง” ตัวละครของปริญญาพนธ์นี้ แสดงดังรูปที่ 3.34



รูปที่ 3.34 ตัวละคร “น้องเทเลคอมจิ้ง”

3) เมื่อออกแบบเสร็จแล้วจึงนำตัวละครมาแยกชิ้นส่วน ได้แก่ แขน 2 ข้าง ขา 2 ข้าง ลำตัว ตา และปาก เพื่อนำไปเข้ากระบวนการ Animation โดยใช้โปรแกรม Clip-Studio Paint

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับการออกแบบและสร้างภาพเคลื่อนไหวให้สอดคล้องกับบทบาทของตัวละคร ท่าทางของตัวละครในปฏิญญาพันธ์ 3 รูปแบบ ได้แก่

3.1) การทักทาย

โดยตัวละครมีท่าทางโบกมือทักทายพร้อมกับเสียงพูดและข้อความว่า “สวัสดีค่ะ มีอะไรจะถามน้องเทเลคอมจิ้งมัยคะ” แสดงดังรูปที่ 3.35



รูปที่ 3.35 ท่าทางทักทายของตัวละคร

3.2) การตอบคำถาม

ตัวละครมีท่าทางขยับปากที่เหมือนกับตัวละครกำลังพูดหรือตอบคำถามที่ผู้ใช้ถามเข้ามา แสดงดังรูปที่ 3.36 และ 3.37



รูปที่ 3.36 ท่าทางขยับเปิดปากของตัวละคร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.37 ท่าทางขยับปิดปากของตัวละคร

3.3) ขณะไม่มีการตอบคำถาม

ตัวละครมีท่าทางขยับตัวขึ้นลงเล็กน้อยและกระพริบตาเพื่อรอให้ผู้ใช้งาน

ถามคำถาม แสดงดังรูปที่ 3.38 และ 3.39



รูปที่ 3.38 ท่าทางของตัวละครที่มีลักษณะเปิดตา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.39 ท่าทางของตัวละครที่มีลักษณะปิดตา

4) เมื่อเสร็จสิ้นขั้นตอนการทำ Animation แล้ว จากนั้นทำการเรนเดอร์ออกมาเป็นภาพที่ไม่มีพื้นหลัง (.png) โดยใช้โปรแกรม Adobe Premiere Pro สำหรับลบพื้นหลังของตัวละคร แสดงดังรูปที่ 3.40 และ 3.41



รูปที่ 3.40 ตัวละครก่อนลบพื้นหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

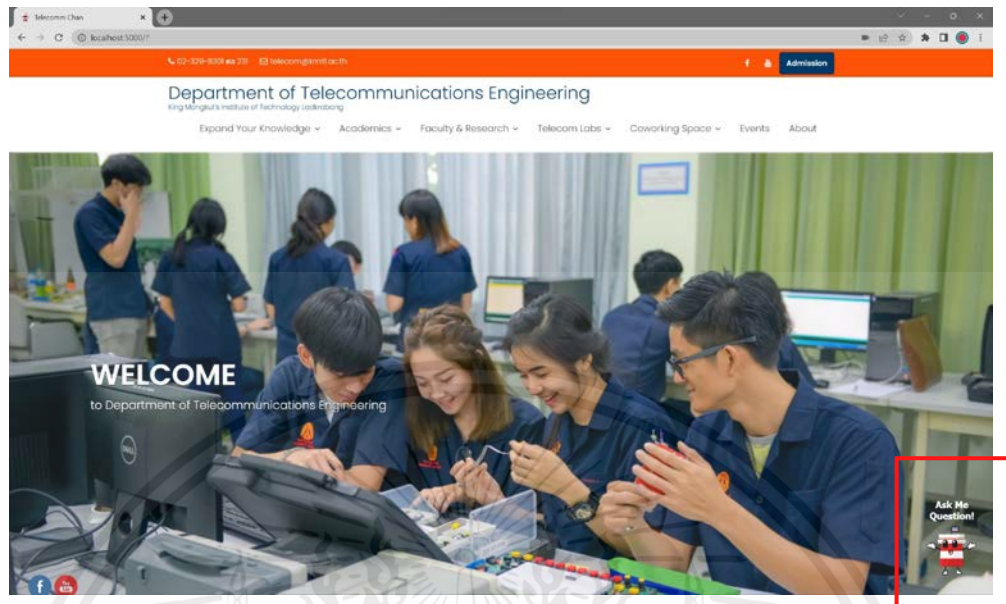


รูปที่ 3.41 ตัวละครหลังลบพื้นหลัง

5) นำภาพทั้งหมดมาเรียงต่อกันในโปรแกรม Adobe Premiere Pro เพื่อเรนเดอร์ออกมาเป็นภาพเคลื่อนไหว (.gif)

3.1.7 การออกแบบหน้าเว็บไซต์สำหรับใช้งาน Recommendation System

สำหรับการออกแบบหน้าเว็บไซต์เพื่อใช้งานระบบ Recommendation System API ในปฏิยานิพนธ์นี้จะจำลองว่ามีการใช้งานบนเว็บไซต์ <http://www.telecom.kmitl.ac.th/> โดยออกแบบให้เป็นหน้าต่าง Chat-Popup และมีปุ่มกดเป็นรูปตัวการ์ตูนพร้อมแสดงข้อความ “Ask Me Question!” ทางด้านล่างขวามือ ดังแสดงในรูปที่ 3.42 และเมื่อเอาตัวชี้ตำแหน่ง (Cursor) ไปชี้ที่ปุ่มจะมีพื้นหลังสีส้มเกิดขึ้น ดังแสดงในรูปที่ 3.43 จากนั้นเมื่อทำการกดปุ่มจะมีหน้าต่าง Chat-Popup ขึ้นมา ดังแสดงในรูปที่ 3.44

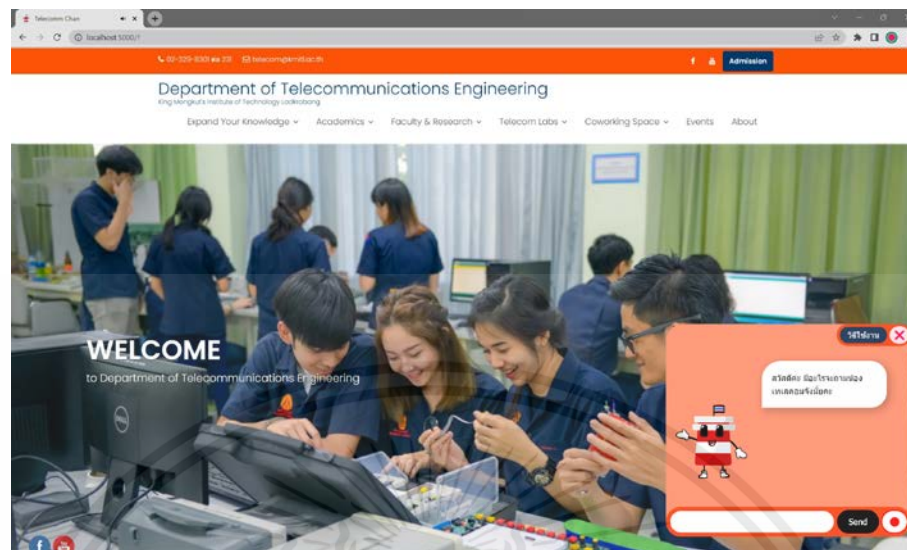


รูปที่ 3.42 หน้าเว็บไซต์ตอนยังไม่กดปุ่ม



รูปที่ 3.43 หน้าเว็บไซต์เมื่อนำตัวชี้ตำแหน่ง (Cursor) ไปชี้ที่ปุ่ม

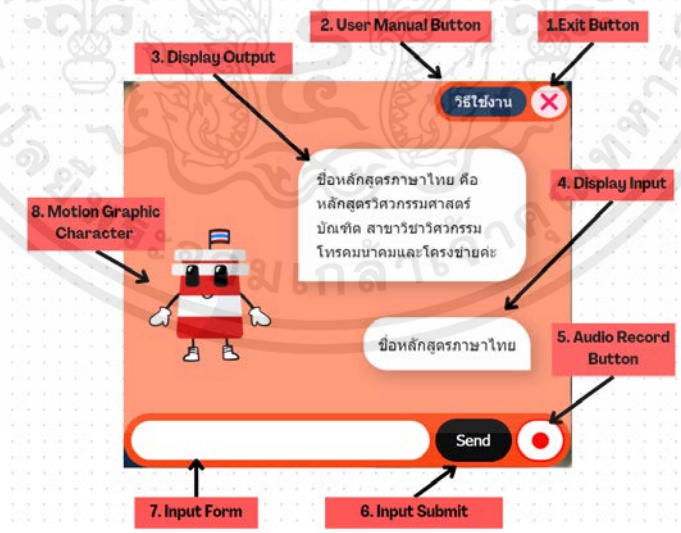
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.44 หน้าเว็บไซต์หลังจากกดปุ่ม

3.1.7.1 การออกแบบส่วนต่อประสานกับผู้ใช้งาน (User Interface: UI)

เมื่อกดปุ่ม “Ask Me Question!” จะมีหน้าต่าง Chat Popup ขึ้นมาให้ใช้งานและมีส่วนประกอบต่าง ๆ ดังแสดงในรูปที่ 3.45



รูปที่ 3.45 ส่วนประกอบต่าง ๆ ของหน้าต่าง Chat-Popup

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.45 แต่ละส่วนจะมีการใช้งาน ดังนี้

1) ปุ่มปิดหน้าต่าง (Exit Button)

ใช้สำหรับออกจากหน้าต่าง Chat-Popup กลับสู่หน้าเว็บไซต์ปกติ โดยออกแบบให้เป็นรูปกากบาทสีแดงวางไว้ที่มุมขวาบนสุดของหน้าต่าง

2) ปุ่มเปิด/ปิดเพื่อดูวิธีการใช้งาน (User Manual Button)

ใช้สำหรับเปิดและปิดคู่มือการใช้งานระบบ โดยออกแบบให้ปุ่มกดแสดงข้อความว่า “วิธีใช้งาน” และมีพื้นหลังสีน้ำเงิน เมื่อทำการกดปุ่มหน้าต่าง Chat-Popup จะแสดงดังรูปที่ 3.46 โดยส่วนแสดงผลเอาต์พุต (Display Output) จะแสดงข้อความที่อธิบายวิธีการใช้งานทั้งหมดแก่ผู้ใช้งาน และมีข้อแนะนำในกรณีที่ผู้ใช้งานไม่สามารถใช้งานปุ่มบันทึกเสียงได้ โดยจะมีแถบเลื่อน (Scroll Bar) ให้ใช้งานด้วย



รูปที่ 3.46 หน้าต่าง Chat-popup แสดงวิธีการใช้งานหลังกดปุ่ม “วิธีใช้งาน”

3) ส่วนแสดงผลเอาต์พุต (Display Output)

ใช้สำหรับแสดงผลข้อความเอาต์พุตทุกอย่าง ทั้งข้อความสำหรับทักทายผู้ใช้งาน ข้อความวิธีการใช้งาน และข้อความคำตอบ โดยออกแบบให้เป็นกล่องข้อความที่มีพื้นหลังสีขาว มีมุมขอบโค้งมนทั้งด้านบนทางซ้าย ด้านบนทางขวา และด้านล่างทางขวา ส่วนด้านล่างทางซ้ายจะเป็นขอบสี่เหลี่ยมเพื่อสื่อความเป็นข้อความที่มาจากตัวการ์ตูนกราฟิกเคลื่อนไหว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) ส่วนแสดงผลอินพุต (Display Input)

ใช้สำหรับแสดงผลข้อความอินพุตที่รับมาจากช่องสำหรับกรอกข้อความ (Input Form) ออกแบบให้มีพื้นหลังสีขาวเช่นเดียวกับส่วนแสดงผลเอาต์พุต มีมุมขอบโค้งมน ด้านบนทางซ้าย ด้านบนทางขวา และด้านล่างทางซ้าย ส่วนด้านล่างทางขวาจะเป็นขอบสีเหลี่ยม เพื่อสื่อความว่าเป็นข้อความที่มาจากตัวผู้ใช้งานเอง

5) ปุ่มสำหรับบันทึกเสียงพูด (Audio Record Button)

ใช้สำหรับการถามคำถามด้วยเสียงพูด หรือกดเพื่อเปิดใช้งาน Speech-to-text API โดยออกแบบให้ปุ่มรูปวงกลมที่สีแดงและขาว เพื่อสื่อความถึงการกดบันทึกเสียง

6) ปุ่มสำหรับส่งอินพุต (Input Submit)

ใช้สำหรับการกดส่งคำถามหรือข้อความของผู้ใช้งานไปยังฝั่งเซิร์ฟเวอร์ (Server-Side) โดยออกแบบให้ปุ่มกลมมีลักษณะเป็นวงรีแสดงข้อความว่า “Send” มีตัวหนังสือสีขาวและพื้นหลังสีดำ

7) ช่องสำหรับกรอกข้อความ (Input Form)

ใช้สำหรับพิมพ์คำถามหรือข้อความ ออกแบบให้มีลักษณะเป็นวงรี เช่นเดียวกับปุ่มสำหรับส่งอินพุตและมีพื้นหลังสีขาว

8) ตัวการ์ตูนกราฟิกเคลื่อนไหว (Motion Graphic Character)

แสดงตัวการ์ตูนกราฟิกเคลื่อนไหวที่มีชื่อว่า “น้องเทเลคอมจิ้ง” ใช้เป็นตัวแทนของระบบเพื่อสื่อสารและทำให้ผู้ใช้งานจดจำได้

3.1.7.2 การออกแบบส่วนของประสบการณ์การใช้งานผู้ใช้ (User Experience: UX)

ในส่วนของประสบการณ์การใช้งานของผู้ใช้ จะมีการออกแบบให้มีความคล้ายคลึงกับ Chat Messenger เพื่อให้ผู้ใช้งานรู้สึกคุ้นเคย และยังมีตัวการ์ตูนกราฟิกเคลื่อนไหวแสดงผลพร้อมเสียงพูด ให้ความรู้สึกเป็นกันเอง โดยจะแบ่งการออกแบบ ดังนี้

1) ออกแบบการทักทายผู้ใช้งาน (Greeting)

เมื่อกดไปที่ปุ่ม “Ask Me Question!” จะมีหน้าต่าง Chat-Popup ขึ้นมา และจะมีประโยคต้อนรับคือ “สวัสดีค่ะ มีอะไรจะถามน้องเทเลคอมจิ้งมั๊ยคะ” พร้อมเสียงพูดและการโบกมือทักทายจากตัวการ์ตูนกราฟิกเคลื่อนไหวน้องเทเลคอมจิ้ง เมื่อตัวการ์ตูนต้อนรับจบ จากนั้นตัวการ์ตูนจะกลับมาอยู่ในสถานะปกติคือจะไม่มีการโบกมือแต่จะกระพริบตาอยู่เรื่อย ๆ ดังแสดงในรูปที่ 3.47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ – หน้าต่าง Chat-Popup หลังกดปุ่ม “Ask Me Question!” (ซ้าย) หน้าต่าง Chat-Popup หลังตัวการ์ตูนพูดทักทายจบ (ขวา)

2) ออกแบบการถามและตอบคำถาม

2.1) การถามคำถามผ่านการช่องสำหรับกรอกข้อความ

เมื่อทำการถามคำถามโดยการพิมพ์เข้าไปในช่องกรอกคำถาม ดังแสดงในรูปที่ 3.48 แล้วกดปุ่ม Send หรือจะกด Enter บนคีย์บอร์ดก็ได้ คำถามที่ถามนั้นจะแสดงขึ้นมาในส่วนแสดงผลอินพุต และระบบจะส่งคำตอบนั้นกลับมาแสดงในส่วนแสดงผลเอาต์พุต และตัวการ์ตูนจะขยับปากตามพร้อมเสียงพูด ดังแสดงในรูปที่ 3.49



รูปที่ 3.48 หน้าต่าง Chat-Popup ในขณะที่มีการถามคำถามในช่องกรอกคำถาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.49 หน้าต่าง Chat-Popup ในขณะที่มีการแสดงคำถามและคำตอบ

2.2) การถามคำถามด้วยเสียงพูด

นอกจากถามคำถามโดยการพิมพ์ในช่องกรอกคำถามแล้ว ผู้ใช้งานยังสามารถถามคำถามด้วยเสียงโดยการกดปุ่มบันทึกเสียงทางด้านล่างขวามือ ดังแสดงในรูปที่ 3.50 ในระหว่างการบันทึกเสียง เมื่อผู้ใช้งานพูด ประโยคที่พูดนั้นจะถูกนำมาแสดงในส่วนของการแสดงผลอินพุตและปุ่มบันทึกจะกลายเป็นรูปสี่เหลี่ยมกระพริบสลับกับสีแดง ดังแสดงในรูปที่ 3.51 เมื่อพูดถามคำถามเสร็จจะต้องกดปุ่มบันทึกเสียงอีกครั้ง เพื่อให้ระบบส่งคำตอบของคำถามนั้นกลับมาแสดงในส่วนแสดงผลเอาต์พุต และตัวการ์ตูนจะขยับปากตามพร้อมเสียงพูด ดังแสดงในรูปที่ 3.52

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.50 ปุ่มบันทึกเสียง



รูปที่ 3.51 หน้าต่าง Chat-Popup ในขณะที่บันทึกเสียง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.52 หน้าต่าง Chat-Popup ในขณะที่มีการแสดงคำถามและคำตอบ

งานได้ 2.3) กรณีที่ Recommendation System API ไม่สามารถใช้

หากผู้ใช้งานทำการถามคำถามและหากระบบของทางฝั่ง Recommendation System API มีการปรับปรุง หรือไม่สามารถเข้าใช้งานได้เนื่องจากกรณีใด ๆ ก็ตาม ระบบจะแสดงข้อความ “ขอภัยในความไม่สะดวก เนื่องจากอยู่ในการปรับปรุงระบบชั่วคราวค่ะ” ที่ส่วนแสดงผลเอาต์พุต ดังแสดงในรูปที่ 3.53



รูปที่ 3.53 หน้าต่าง Chat-Popup กรณีที่ Recommendation System API ไม่สามารถใช้งานได้

3.1.7.3 การออกแบบเว็บไซต์สำหรับขนาดหน้าจอต่าง ๆ (Responsive Web Design)

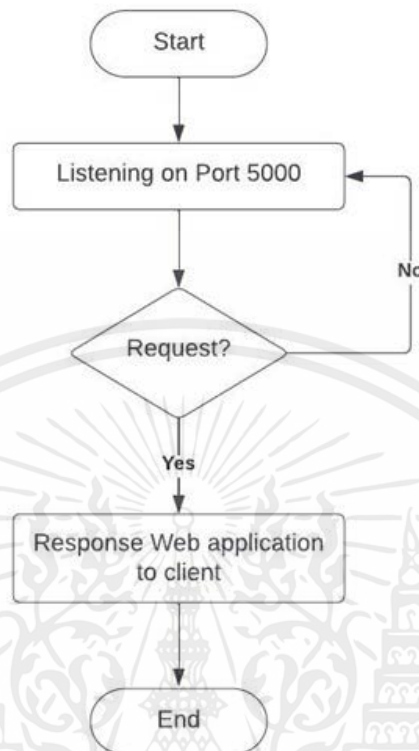
ในปฏิญญาฉบับนี้จะเน้นการรองรับการใช้งานผ่านเว็บเบราว์เซอร์บนเครื่องเดสก์ท็อปเป็นหลัก โดยจะออกแบบการรองรับขนาดของหน้าจอต่าง ๆ ตั้งแต่ขนาดความกว้าง 1366px เป็นต้นไปและจะใช้งานได้ดีที่สุดในขนาด 1920px เนื่องจากเป็นขนาดที่นิยมมากที่สุด ตามสถิติขนาดหน้าจอเว็บไซต์ที่ใช้มากที่สุดในประเทศไทย อ้างอิงจาก <https://gs.statcounter.com/screen-resolution-stats/desktop/thailand>

นอกจากนี้ยังออกแบบให้สามารถใช้งานบนโทรศัพท์มือถือ และแท็บเล็ตได้อีกด้วย โดยในโทรศัพท์จะใช้งานได้ที่หน้าจอขนาดความกว้าง 320px - 768px และในแท็บเล็ตจะใช้งานได้ที่ขนาด 769px - 1365px

3.1.8 การออกแบบเว็บเซิร์ฟเวอร์

การออกแบบเว็บเซิร์ฟเวอร์สำหรับบริการเว็บแอปพลิเคชันที่ได้ทำการออกแบบไปในหัวข้อที่ 3.1.7 มีการทำงานดังแสดงในรูป 3.54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.54 บล็อกไดอะแกรมแสดงการทำงานของเว็บเซิร์ฟเวอร์

3.1.8.1 การสร้างโปรเจค Node.js สำหรับสร้างเว็บเซิร์ฟเวอร์

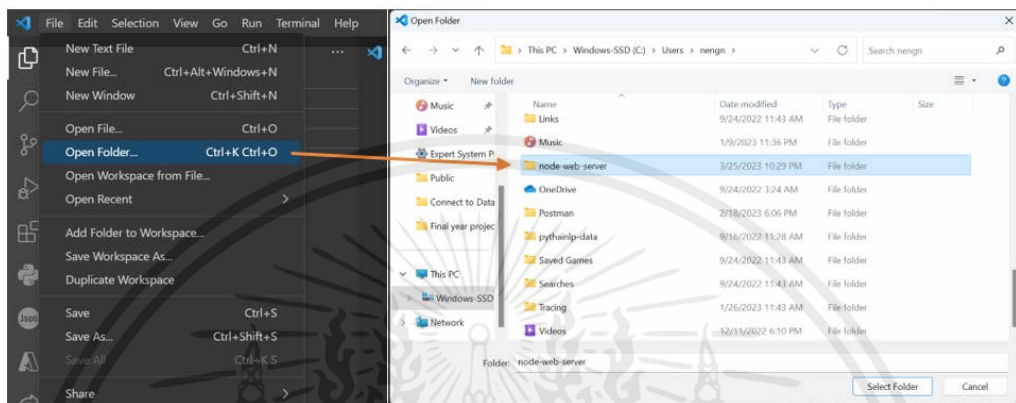
ให้ทำการเปิด Command Prompt ขึ้นมาและสร้างสสารบบ (Directory) หรือโฟลเดอร์ใหม่ที่ชื่อว่า node-web-server ด้วยคำว่า `mkdir node-web-server` ดังแสดงในรูปที่ 3.55

```
C:\Users\nengn>mkdir node-web-server|
```

รูปที่ 3.55 การสร้างสสารบบใหม่สำหรับโปรเจค Node.js

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นให้เปิด Visual Studio Code ขึ้นมา และเข้าไปที่ File -> Open Folder แล้วให้เลือกโฟลเดอร์ node-web-server ดังแสดงในรูปที่ 3.56 จากนั้นให้ทำการสร้างไฟล์ package.json โดยพิมพ์คำสั่ง `npm init -y` ใน terminal ดังแสดงรูปที่ 3.57



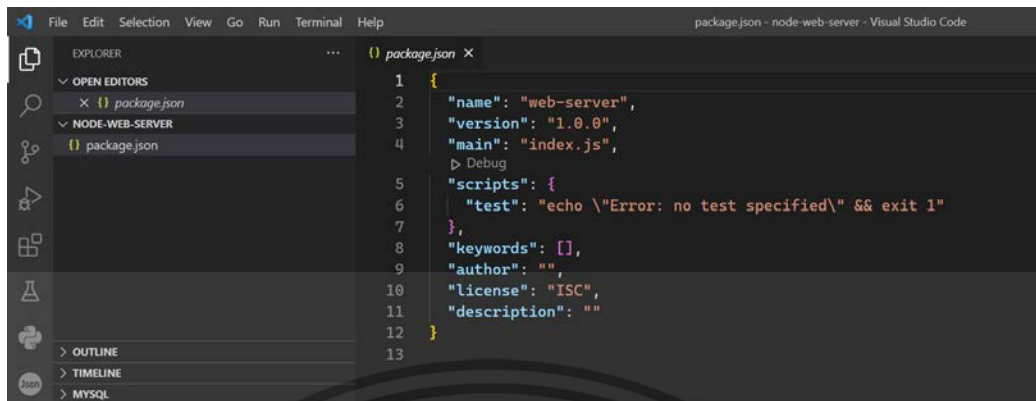
รูปที่ 3.56 การสร้างโปรเจกใหม่ในโฟลเดอร์ node-web-server



รูปที่ 3.57 สร้างไฟล์ package.json

จะเห็นไฟล์ package.json ถูกสร้างมาขึ้นในโฟลเดอร์ node-web-server ดังแสดงในรูปที่ 3.58 โดย package.json เป็นไฟล์ที่ใช้สำหรับอธิบายรายละเอียดเกี่ยวกับโปรเจก โดยจะเก็บรายชื่อของแพ็คเกจต่าง ๆ ที่ใช้งานในโปรเจก ซึ่งเป็นไฟล์ที่ใช้บอกกับ npm (Node Package Management) ในการค้นหาและติดตั้งแพ็คเกจให้กับโปรเจกที่จะทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

1  {
2    "name": "web-server",
3    "version": "1.0.0",
4    "main": "index.js",
5    "scripts": {
6      "test": "echo \\\"Error: no test specified\\\" && exit 1"
7    },
8    "keywords": [],
9    "author": "",
10   "license": "ISC",
11   "description": ""
12 }
13

```

รูปที่ 3.58 ไฟล์ package.json

ถัดมาจะทำการติดตั้งแพ็คเกจหรือ Framework ที่จำเป็นคือ Express และ Socket.io โดยใช้คำสั่ง `npm install express socket.io` ไปที่ Terminal ดังแสดงในรูปที่ 3.59 เมื่อสังเกตในไฟล์ package.json ใน key ของ dependencies จะมี value ของ express และ socket.io เพิ่มขึ้นและมีไฟล์ package-lock.json ถูกสร้างขึ้นมาด้วย ดังแสดงในรูปที่ 3.60



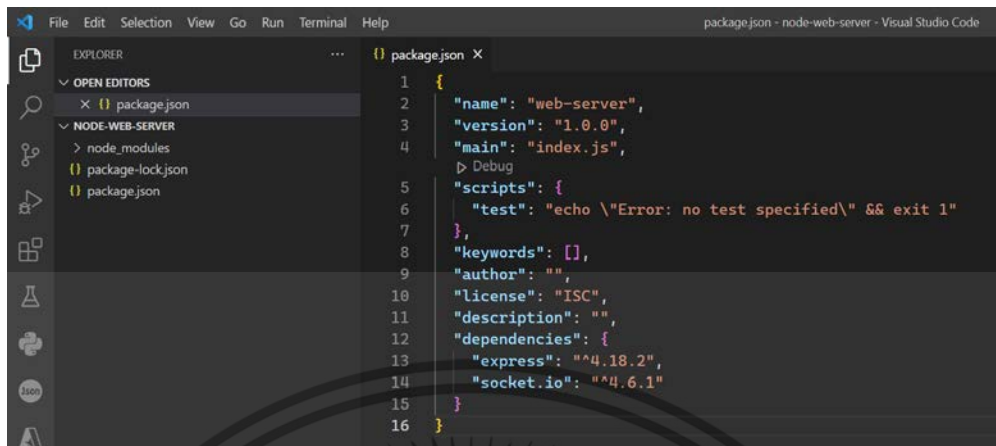
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE
PS C:\Users\nengn\node-web-server> npm install express socket.io

```

รูปที่ 3.59 การติดตั้งแพ็คเกจ Express และ Socket.io

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



```

1  {
2    "name": "web-server",
3    "version": "1.0.0",
4    "main": "index.js",
5    "scripts": {
6      "test": "echo \\\"Error: no test specified\\\" && exit 1"
7    },
8    "keywords": [],
9    "author": "",
10   "license": "ISC",
11   "description": "",
12   "dependencies": {
13     "express": "^4.18.2",
14     "socket.io": "^4.6.1"
15   }
16 }

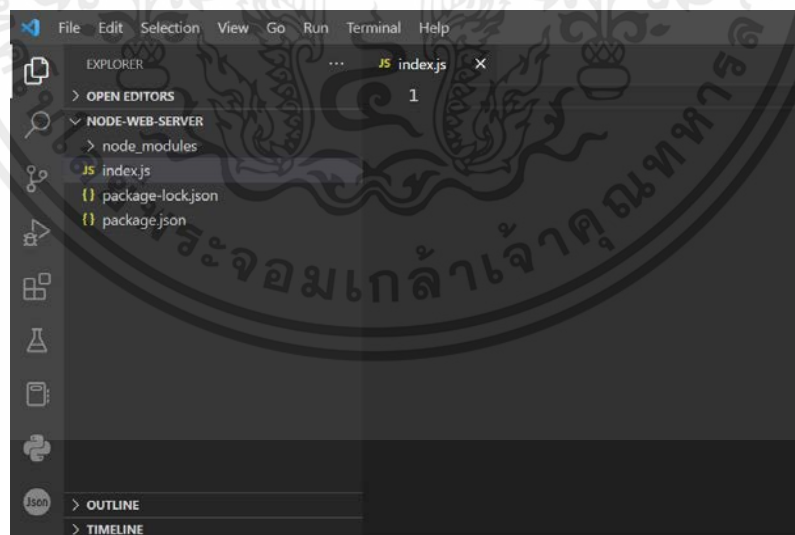
```

รูปที่ 3.60 ข้อมูลในไฟล์ package.json เมื่อติดตั้ง Express และ Socket.io

3.1.8.2 สร้างเว็บเซิร์ฟเวอร์ ด้วย Express และ Socket.io

หลังจากที่ได้สร้างโปรเจกต์ Node.js แล้ว ต่อไปจะทำการสร้างเว็บเซิร์ฟเวอร์สำหรับใช้งาน Recommendation System โดยมีขั้นตอนดังนี้

1) ทำการสร้างไฟล์ index.js เพิ่มในโฟลเดอร์ node-web-server เพิ่มเขียนโปรแกรมรับคำถามจากผู้ใช้งานจากหัวข้อที่ 3.2 ดังแสดงในรูปที่ 3.61



```

1

```

รูปที่ 3.61 ไฟล์ index.js สำหรับโปรแกรมสร้าง Server

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

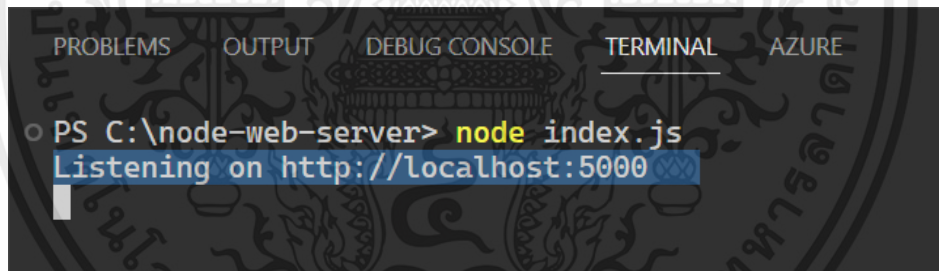
2) จากนั้นจะทำการสร้างเซิร์ฟเวอร์ที่ `http://localhost:5000/` แล้วสั่งให้แสดงข้อความ “Listening on `http://localhost:5000`” ผ่าน Terminal ด้วยคำสั่ง `Console.log ()` โดยการรันโปรแกรมด้วยคอมมานด์ “`node index.js`” ดังแสดงในรูปที่ 3.62 และแสดงผลลัพธ์ดังแสดงในรูปที่ 3.63

```

1  const express = require('express');
2  const app = express();
3  const http = require('http');
4  const server = http.createServer(app);
5  const { Server } = require("socket.io"); 293.1k (gzipped: 62.6k)
6  const PORT = 5000;
7
8  server.listen(PORT, () => {
9    console.log(`Listening on http://localhost:${PORT}`)
10 });
11

```

รูปที่ 3.62 โปรแกรมฝั่ง Server



The screenshot shows a terminal window with the following content:

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  AZURE
PS C:\node-web-server> node index.js
Listening on http://localhost:5000

```

รูปที่ 3.63 เปิดพอร์ต (Port) รอการเชื่อมต่อที่เบอร์ 5000

3) เมื่อเซิร์ฟเวอร์เปิดพอร์ตที่ 5000 แล้ว ถัดมาจะสร้าง Routing ให้ผู้ใช้งานเว็บไซต์มีการร้องขอด้วย Method GET (GET Request) ด้วย URL “`http://localhost:5000/`” ดังแสดงในรูปที่ 3.64

```
app.get('/', (req, res) => {
  // Response to client-browser
});
```

รูปที่ 3.64 การตอบสนองของเซิร์ฟเวอร์เมื่อมี GET Request ที่ URL http://localhost:5000/

4) ทำการรวม Socket.io เข้ากับ HTTP Server ที่สร้างขึ้นด้วยการสร้างอ็อบเจกต์จากคลาส Server และทำการเปิดโพรโทคอล Web Socket ด้วยการเรียกใช้เมทอด io.on() ดังแสดงในรูปที่ 3.65

```
const io = new Server(server);
io.on('connection', (socket) => {
  //Do something...
});
```

รูปที่ 3.65 การเชื่อมต่อของฝั่งผู้ใช้บริการและผู้ให้บริการ

จากนั้นหากต้องการส่งข้อมูลจากผู้ให้บริการไปยังผู้ใช้บริการจะสามารถทำได้โดยใช้คำสั่ง socket.emit() ผ่านเหตุการณ์ (Event) “new-messages” และหากต้องการรับข้อมูลจากผู้ให้บริการทำได้โดยใช้คำสั่ง socket.on() ผ่านเหตุการณ์ (Event) “messages” ดังแสดงในรูปที่ 3.66

```
const io = new Server(server);
Complexity is 3 Everything is cool!
io.on('connection', (socket) => {
  socket.on('messages', () => { /*Received request from Client*/ });
  socket.emit('new-message', () => { /*response data to Client*/ });
});
```

รูปที่ 3.66 คำสั่ง socket.on() และ socket.emit()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.8.3 การเรียกใช้ Recommendation System API บน Node.js

กรณีที่มีคำถามจากผู้ใช้งานคำถามนี้จะถูกส่งต่อไปที่ Recommendation System API โดยใช้ฟังก์ชัน fetch() โดยมี URL คือ `https://telecom-qa-app-oio6zs2epa-as.a.run.app/predict` และจะต้องปรับแต่งข้อมูลต่าง ๆ ลงในตัวอ็อบเจกต์ options ดังแสดงในรูปที่ 3.67 โดย HTTP Request ที่ใช้คือเมทอด POST เพราะใช้ในการเพิ่มข้อมูลเข้าไปที่ API และในส่วน Headers จะกำหนด Content-Type คือ `application/json` เพื่อบอกว่าข้อมูลที่ส่งไปนั้นเป็นชนิด JSON สุดท้ายคือ body เป็นส่วนที่บรรจุข้อมูลที่ต้องการส่งไป ซึ่งโดยปกติแล้วนั้นทางผู้บริการ API จะต้องแจ้งการปรับแต่งค่าต่าง ๆ ในอ็อบเจกต์ options

```
const url = 'https://telecom-qa-app-oio6zs2epa-as.a.run.app/predict';
const options = {
  method: 'POST',
  headers: { 'content-type': 'application/json' },
  body: '{ "text": "ชื่อหลักสูตรคืออะไร" }'
};
```

รูปที่ 3.67 URL และข้อมูลของอ็อบเจกต์ options ที่ใช้ในการ fetch() API

เมื่อได้ url และ options จะส่งคำขอ (Request) ไปยัง API ด้วยคำสั่ง `Fetch(url, options)` และจะรับ Response และจะทำการแปลงข้อมูลให้เป็น JSON ด้วยคำสั่ง `.then(res => res.json())` จากนั้นจะเข้าถึงข้อมูลนี้ด้วยคำสั่ง `.then(json => console.log('Answer: '+json.answer))` และสุดท้ายจะตรวจสอบความผิดพลาดในกรณีส่ง HTTP Request ไม่สำเร็จด้วยคำสั่ง `.catch(err => console.error('error: ' + err))` ดังแสดงในรูปที่ 3.68

```
fetch(url, options)
  .then(res => res.json())
  .then(json => console.log('Answer: ' + json.answer))
  .catch(err => console.error('error: ' + err))
```

รูปที่ 3.68 คำสั่ง fetch()

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.8.4 การเรียกใช้ Google Cloud Platform API บน Node.js

ในการใช้งาน Recommendation System ที่เว็บไซต์ นอกจากจะแสดงผลด้วยข้อความแล้ว ยังสามารถแสดงผลด้วยเสียงด้วย จึงได้ออกแบบให้ระบบมีการใช้งาน Text-to-speech โดยจะใช้บริการ Cloud Text-to-speech API ของ Google Enterprise API มาใช้งานในปฏิญานิพนธ์นี้ โดยการนำ Cloud Text-to-speech API มาใช้งาน จะมีขั้นตอนดังต่อไปนี้

1) ติดตั้งแพ็คเกจสำหรับการใช้งานบริการ Cloud Text-to-speech API ให้ทำการเปิด Command Prompt ขึ้นมาและไปยัง Directory ของโปรเจค ด้วยคำสั่ง `cd` ชื่อ Directory ของโปรเจค จากนั้นให้ทำการติดตั้งแพ็คเกจด้วยคำสั่ง `npm install @google-cloud/text-to-speech` ดังแสดงในรูปที่ 3.69

```
C:\Users\nengn>cd node-web-server
C:\Users\nengn\node-web-server>npm install @google-cloud/text-to-speech|
```

รูปที่ 3.69 การติดตั้งแพ็คเกจ @google-cloud/text-to-speech

2) การใช้งาน Cloud Text-to-speech API บน Node.js
เมื่อทำการติดตั้งแพ็คเกจเรียบร้อยแล้ว จะทำการเรียกใช้โดยมานำเก็บไว้ในตัวแปร `textToSpeech` และเข้าถึงเมทอด `TextToSpeechClient` โดยสร้างอ็อบเจกต์ Client มารองรับ ดังแสดงในรูปที่ 3.70

```
const textToSpeech = require('@google-cloud/text-to-speech');
const client = new textToSpeech.TextToSpeechClient();
```

รูปที่ 3.70 การเรียกใช้แพ็คเกจ @google-cloud/text-to-speech

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อมาจะทำการเรียกใช้งาน API นี้ โดยจะสร้างตัวแปร request ซึ่งจะมีการปรับแต่งข้อมูลดังนี้

1) input: ข้อความที่ต้องการนำไปทำ Text-to-speech
 2) voice: การเลือกภาษาที่ใช้อ่านออกเสียง หรือ languageCode และเสียงของ Text-to-speech หรือ ssmlGender โดยในปฏิญานิพนธ์นี้จะเลือก languageCode: th-TH และ ssmlGender: FEMALE

3) audioconfig: การปรับแต่งข้อมูลเสียง ดังนี้ audioEncoding คือการเลือกชนิดการถอดรหัสข้อมูลเสียงนี้ โดยมีให้เลือกคือ MP3 และ LINEAR16, pitch คือการปรับแต่งโทนเสียง และ speakRate คือความเร็วในการอ่านออกเสียง โดยในปฏิญานิพนธ์นี้จะเลือกใช้ audioEncoding: 'MP3', pitch: 5 และ speakRate: 1.05

จากนั้นตัวแปร request นี้จะถูกส่งไปที่ API ด้วยเมทอด synthesizeSpeech() แล้วจะได้รับการตอบกลับมาที่ตัวแปร response ซึ่งเป็น arraybuffer จากนั้นจะทำการเข้าถึงข้อมูลเสียงแล้วเก็บไว้ที่ตัวแปร audioContent ด้วยคำสั่ง response.audioContent ดังแสดงในรูปที่ 3.71

```
const textToSpeech = require('@google-cloud/text-to-speech');
const client = new textToSpeech.TextToSpeechClient();

const request = {
  input: { text: /*'text'*/ },
  voice: { languageCode: 'th-TH', ssmlGender: 'FEMALE' },
  audioConfig: {
    audioEncoding: 'MP3',
    pitch: 5,
    speakingRate: 1.05
  },
};

const [response] = await client.synthesizeSpeech(request);
const audioContent = response.audioContent;
```

รูปที่ 3.71 การใช้งาน Cloud Text-to-speech API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.9 การสื่อสารระหว่างฝั่งผู้ใช้งานกับระบบหลังบ้านทั้งหมด (Back-end)

ในการสื่อสารระหว่างผู้ใช้งานกับระบบหลังบ้าน (Back-end) จะสร้างโปรแกรมฝั่งผู้ใช้งาน (Client) ในไฟล์ที่ชื่อว่า index.html ด้วย HTML5 ดังแสดงในรูปที่ 3.72 โดยจะต้องรองรับการใช้งานการรับ-ส่งข้อความ การเปิดไฟล์เสียง และการบันทึกเสียงของผู้ใช้ หรือ Speech-to-text



```

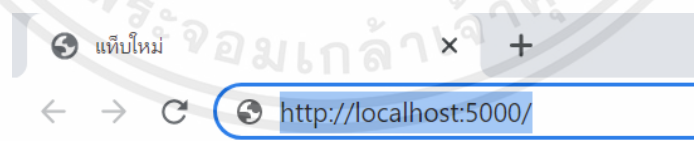
index.html X
1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5   <meta charset="UTF-8">
6   <meta http-equiv="X-UA-Compatible" content="IE=edge">
7   <meta name="viewport" content="width=device-width, initial-scale=1.0">
8   <title>Document</title>
9 </head>
10
11 <body>
12
13 </body>
14
15 </html>

```

รูปที่ 3.72 โปรแกรมฝั่งผู้ใช้งาน (Client)

1) การร้องขอหน้าเว็บไซต์จากเว็บเซิร์ฟเวอร์ (Request Webpage)

ในขั้นตอนแรกของการใช้งานนั้น ผู้ใช้งานจำเป็นต้องส่งคำขอ (Request) หน้าเว็บไซต์ ด้วย HTTP Method GET โดยพิมพ์ URL `http://localhost:5000/` ตามที่ได้สร้างขึ้นในหัวข้อที่ 3.1.8.2 ที่ Web Browser ดังแสดงในรูปที่ 3.73



รูปที่ 3.73 ผู้ใช้งานส่งคำขอ (Request) มาที่เว็บเซิร์ฟเวอร์ `http://localhost:5000/`

2) การออกแบบการส่ง-รับข้อมูลระหว่างผู้ใช้งานกับเว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะโปรแกรมสร้างส่วนใช้งานฝั่ง Client ได้ดังแสดงในรูปที่ 3.74 ก่อนอื่นจะต้องติดตั้ง socket.io ที่ฝั่ง Client ก่อน ด้วยวิธีแบบ CDN (Content Delivery Network) จากนั้นจะสร้าง HTML Tag Form สำหรับรับข้อความจากผู้ใช้ เมื่อทำการ Submit โดยการกดปุ่ม Send ข้อความนี้จะถูกส่งไปที่เว็บเซิร์ฟเวอร์ ผ่านเหตุการณ์ (Event) 'messages' ด้วยคำสั่ง client.emit()

```

10
11 <body>
12   <form>
13     <input>
14     <button>Send</button>
15   </form>
16
17   <script src="/socket.io/socket.io.js"></script>
18   <script>
19     const client = io();
20     const input = document.querySelector('input')
21     const form = document.querySelector('form')
22
23     form.addEventListener('submit', (e) => {
24       e.preventDefault();
25       client.emit('messages', input.value);
26     })
27   </script>
28 </body>
29

```

รูปที่ 3.74 โปรแกรมฝั่งผู้ใช้งานในการส่งข้อมูลไปที่เว็บเซิร์ฟเวอร์

จากนั้นจะทำการโปรแกรมส่วนการรับการตอบสนอง (Response) จากเว็บเซิร์ฟเวอร์ ด้วยคำสั่ง client.on() ผ่านเหตุการณ์ "new-message" และนำข้อมูลที่รับได้มาเก็บไว้ในตัวแปร res_messages จากนั้นนำข้อมูลไปแสดงผลใน HTML Tag div ดังแสดงในรูปที่ 3.75

```

<body>
  <!-- Request to Server-->
  <form>
    <input>
    <button>Send</button>
  </form>

  <!-- Response from Server-->
  <div></div>

  <script src="/socket.io/socket.io.js"></script>
  <script>
    const client = io();
    const input = document.querySelector('input');
    const form = document.querySelector('form');
    form.addEventListener('submit', (e) => {
      e.preventDefault();
      client.emit('messages', input.value);
    });
    client.on('new-message', (res_messages) => {
      document.querySelector('div').innerHTML = res_messages;
    });
  </script>
</body>
</html>

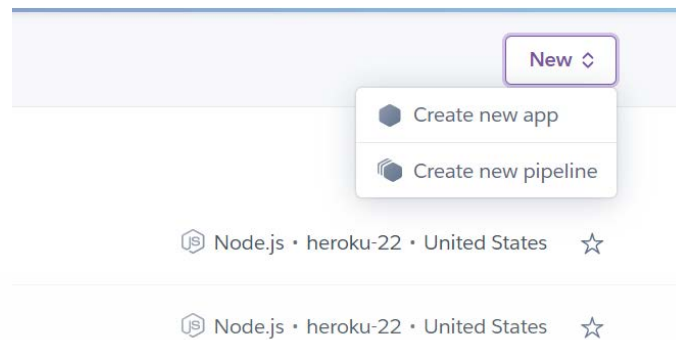
```

รูปที่ 3.75 โปรแกรมฝั่งผู้ใช้งาน (Client) ในการรับข้อมูลจากเว็บเซิร์ฟเวอร์

3.1.10 การ Deploy เว็บเซิร์ฟเวอร์ ไปบน Heroku

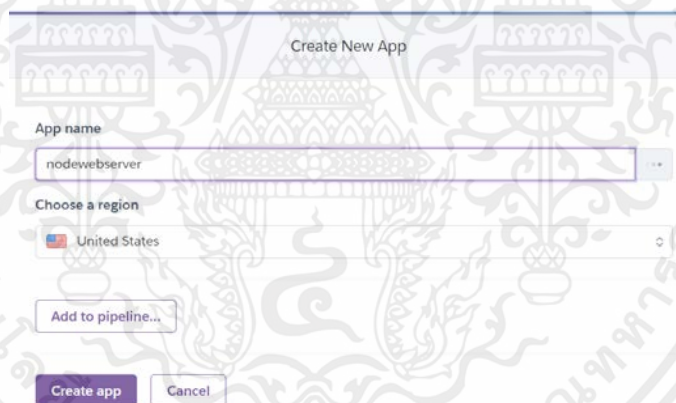
หลังจากที่ได้สร้างเว็บเซิร์ฟเวอร์ ที่พร้อมใช้งานแล้ว ขั้นตอนที่สำคัญถัดมาคือการเปิดให้ใช้งานสู่สาธารณะ โดยการ Deploy ไปบน Heroku ซึ่งเป็นบริการคลาวด์แบบ Platform as a Service (PaaS) เจ้าหนึ่ง โดยมีการ Deploy ไปบน Heroku นั้น ต้องทำการติดตั้ง git ลงในเครื่องคอมพิวเตอร์และสมัครสมาชิกให้เรียบร้อยก่อน หลังจากนั้น จะมีขั้นตอนการ Deploy ดังนี้

1) เข้าสู่เว็บไซต์ <https://dashboard.heroku.com/apps> จะทำการสร้างแอปพลิเคชันใหม่ โดยให้คลิกปุ่ม New และเลือก Create new app ดังแสดงในรูปที่ 3.76



รูปที่ 3.76 การสร้างแอปพลิเคชันใหม่บน Heroku

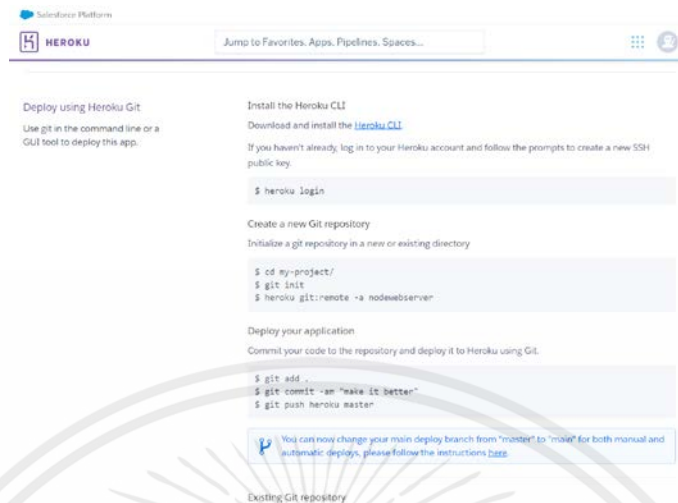
2) ให้ทำการตั้งชื่อแอปพลิเคชันในช่อง App name โดยต้องเป็นชื่อภาษาอังกฤษ ตัวพิมพ์เล็กหรือตัวเลขและห้ามมีอักขระพิเศษ ดังแสดงในรูปที่ 3.77 หลังจากนั้นให้กดปุ่ม Create app



รูปที่ 3.77 การตั้งชื่อแอปพลิเคชัน

3) หลังจากนั้นให้ดำเนินการ Deploy เว็บเซิร์ฟเวอร์ ตามขั้นตอนที่เว็บไซต์ของ Heroku แนะนำ ดังแสดงในรูปที่ 3.78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.78 ขั้นตอนการ Deploy เว็บเซิร์ฟเวอร์

3.2 อุปกรณ์ที่ใช้ในการทดลอง

ในโครงงานนี้ มีอุปกรณ์และเครื่องมือที่ใช้ในการทดลอง ดังนี้

3.2.1 Google Colab Pro

Google Colaboratory หรือ Colab เป็นบริการคลาวด์อีกหนึ่งบริการจาก Google Research เป็น IDE ที่อนุญาตให้ผู้ใช้เขียนซอร์สโค้ดในตัวแก้ไขและเรียกใช้จากเบราว์เซอร์ โดยเฉพาะอย่างยิ่ง รองรับภาษาการเขียนโปรแกรม Python และเน้นงานการเรียนรู้เครื่อง การวิเคราะห์ข้อมูล โครงการการศึกษา ฯลฯ [62]

3.2.2 Visual Studio Code

เป็นโปรแกรมประเภท Editor ใช้ในการแก้ไขโค้ดที่มีขนาดเล็ก แต่มีประสิทธิภาพสูง เป็น Open Source โปรแกรมจึงสามารถนำมาใช้งานได้โดยไม่มีค่าใช้จ่าย เหมาะสำหรับนักพัฒนา โปรแกรมที่ต้องการใช้งานหลายแพลตฟอร์ม รองรับการใช้งานทั้งบน Windows, macOS และ Linux รองรับหลายภาษาทั้ง JavaScript, TypeScript และ Node.js ในตัว และสามารถเชื่อมต่อกับ Git ได้ง่าย สามารถนำมาใช้งานได้ง่ายไม่ซับซ้อน มีเครื่องมือและส่วนขยายต่าง ๆ ให้เลือกใช้มากมาย รองรับการทำงานภาษาอื่น ๆ ทั้ง ภาษา C++, C#, Java, Python และ PHP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.3 Jupyter Notebook

Jupyter Notebook เป็นโปรแกรมแบบ Open Source ที่ใช้สำหรับเขียนและทดสอบโค้ด (code) ในภาษาโปรแกรมต่าง ๆ จึงนำมาใช้เป็นเครื่องมือสำหรับเขียนคำสั่งภาษา Python สำหรับทดสอบการทำงานของแบบจำลอง

3.2.4 โปรแกรม Postman

Postman เป็นเครื่องมือที่ใช้สำหรับทดสอบและจำลองการทำงานของ API ซึ่งเป็นช่องทางในการสื่อสารระหว่างแอปพลิเคชันต่าง ๆ หรือระบบต่าง ๆ ในรูปแบบของ Request และ Response โดย Postman ช่วยให้ผู้ใช้สามารถทดสอบ API ได้ง่ายขึ้น และสามารถเช็คความถูกต้องของการตอบกลับของ API แต่ละแห่งได้อย่างรวดเร็ว

3.2.5 Node.js

Node.js มีไว้เพื่อใช้เขียนโปรแกรมด้วยภาษา JavaScript โดย Node.js สามารถทำงานได้ในส่วนของ Server นอกจากนี้ Node.js ยังมีการพัฒนา Framework ต่าง ๆ อย่าง Express.js ที่ช่วยให้การพัฒนาเว็บแอปพลิเคชันด้วย Node.js ง่ายขึ้นและมีความยืดหยุ่นในการทำงาน

3.2.6 Google Cloud Platform

Google Cloud Platform เป็นเครื่องมือที่ใช้ในการ Deploy ระบบต่าง ๆ ซึ่ง Google Cloud Platform เป็นบริการคลาวด์ (Cloud Service) ที่ให้บริการโซลูชันคลาวด์ต่าง ๆ อย่างเช่นการเก็บข้อมูล (Storage), เครื่องเสมือน (Virtual Machine) และบริการอื่น ๆ ที่เกี่ยวข้องกับคลาวด์ โดยเฉพาะเพื่อการพัฒนาแอปพลิเคชัน (Application) ที่ใช้งานได้ตลอดเวลา

3.2.7 Docker

Docker เป็นแพลตฟอร์มสำหรับจัดการและเรียกใช้งานแอปพลิเคชันในรูปแบบของคอนเทนเนอร์ ซึ่งเป็นเทคโนโลยีที่ช่วยให้สามารถสร้างและรันแอปพลิเคชันในสภาพแวดล้อมเดียวกันได้ทั้งบนเครื่อง PC หรือบนคลาวด์ (Cloud) โดยที่ไม่ต้องกังวลเรื่องความแตกต่างของเครื่องหรือระบบปฏิบัติการ

3.2.8 MySQL

MySQL เป็นระบบจัดการฐานข้อมูลแบบ Relational Database Management System (RDBMS) ที่มีเป้าหมายเพื่อใช้สำหรับการจัดเก็บและจัดการข้อมูลฐานข้อมูล มีคำสั่ง SQL (Structured Query Language) เพื่อใช้ในการค้นหาและเรียกใช้ข้อมูลที่เก็บอยู่ในฐานข้อมูล อีกทั้งยังมีฟังก์ชันการจัดการข้อมูลที่หลากหลาย เช่น เพิ่ม, ลบ, แก้ไขข้อมูล รวมถึงการจัดการกับตาราง

(Table) และฟิลด์ (Field) ซึ่งจะช่วยให้ผู้ใช้งานสามารถจัดการข้อมูลฐานข้อมูลได้อย่างมีประสิทธิภาพและง่าย

3.2.9 Adobe Premiere Pro

เป็นโปรแกรมตัดต่อที่มีความสามารถหลากหลาย เช่น การตัดต่อคลิปวิดีโอ, การลบพื้นหลังของวิดีโอ, การเพิ่มเอฟเฟกต์และกราฟิกเคลื่อนไหว, การสร้างภาพนิ่งจากวิดีโอ, และอื่นๆ

3.2.10 Clip – Studio Paint

เป็นโปรแกรมสำหรับวาดรูปและตัดต่อภาพ Clip – Studio Paint มีเครื่องมือให้เลือกหลากหลายชนิด เช่น การวาดสีธรรมชาติ, การวาดและปรับแต่งเส้นต่างๆ และยังมีเครื่องมือสำหรับการตัดต่อภาพ เช่น การตัดต่อและเพิ่มเอฟเฟกต์, การสร้างภาพเคลื่อนไหว เพื่อให้ผู้ใช้งานสามารถสร้างงานได้อย่างมีคุณภาพ

3.3 การจัดเก็บผลการทดลอง

3.3.1 ทดสอบและประเมินประสิทธิภาพของ Reader Model

3.3.1.1 ทดสอบกระบวนการเตรียมข้อมูล (Pre-processing)

3.3.1.2 ทดสอบกระบวนการหาพารามิเตอร์ที่ดีที่สุด (Hyperparameter Optimization)

3.3.1.3 ประเมินประสิทธิภาพของแบบจำลองด้วยผลคะแนน Exact Match และ F1

3.3.2 ทดสอบการใช้งาน Knowledge Base

3.3.2.1 ทดสอบการเรียกดูข้อมูลบริบทจากตารางสำหรับเก็บข้อมูลบริบท

3.3.2.2 ทดสอบการเพิ่มคำถามไปที่ตารางสำหรับคำถามที่ระบบไม่สามารถตอบได้

3.3.3 ทดสอบระบบ Recommendation System API

3.3.3.1 ทดสอบระบบตรวจสอบการทักทาย

3.3.3.2 ทดสอบระบบเตรียมข้อมูลคำถาม (Question Preprocessing)

3.3.3.3 ทดสอบระบบเตรียมข้อมูลบริบท (Contexts Preprocessing)

3.3.3.4 ทดสอบระบบค้นคืนข้อมูล (Retriever)

3.3.3.5 ทดสอบการใช้งาน Reader Model

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.3.6 ทดสอบการเชื่อมต่อ Recommendation System API ด้วยโปรแกรม Postman

3.3.4 ทดสอบการสื่อสารระหว่างเว็บเซิร์ฟเวอร์ กับ API

3.3.4.1 ทดสอบการเชื่อมต่อกับเว็บเซิร์ฟเวอร์ ด้วยโปรแกรม Postman

3.3.4.2 ทดสอบการส่ง-รับข้อมูลที่เว็บเซิร์ฟเวอร์ กับ Recommendation System API

3.3.4.3 ทดสอบการส่ง-รับข้อมูลที่เว็บเซิร์ฟเวอร์ กับ Google Cloud Text-to-speech API

3.3.5 ทดสอบการสื่อสารระหว่างฝั่งผู้ใช้งาน (Client) กับระบบหลังบ้านทั้งหมด (Back-end)

3.3.5.1 ทดสอบการร้องของหน้าเว็บไซต์ด้วย Web browser

3.3.5.2 ทดสอบการส่ง-รับข้อมูลระหว่าง Client กับเว็บเซิร์ฟเวอร์

3.3.6 ทดสอบการใช้งานเว็บไซต์ที่ขนาดหน้าจอต่างกัน (Responsive Web Design)

3.3.7 ทดสอบการใช้งานระบบรวม

บทที่ 4

ผลการทดลอง

ผู้จัดทำได้ทำการเก็บผลการทำงานของระบบโดยแบ่งการทดลองและจัดเก็บผลการทดลองเป็นส่วนต่างๆ ดังต่อไปนี้

4.1 ทดสอบและประเมินประสิทธิภาพของ Reader Model

4.1.1 ทดสอบกระบวนการเตรียมข้อมูล (Pre-processing)

สำหรับการทดสอบกระบวนการเตรียมข้อมูล จะใช้ชุดข้อมูลดังแสดงในรูปที่ 4.1

```
1 example = {
2   "context": "พลอยอยู่ปี 4 กำลังเรียนที่วิศวะลาดกระบัง",
3   "question": "พลอยเรียนที่ไหน",
4   "answers": {
5     "text": ["เรียนที่วิศวะลาดกระบัง"],
6     "answer_start": [18],
7     "answer_end": [40],
8   }
9 }
```

รูปที่ 4.1 ข้อมูลตัวอย่างสำหรับการทดสอบกระบวนการเตรียมข้อมูล

4.1.1.1 ทดสอบกระบวนการ Tokenization ด้วยการใช้ AutoTokenizer

Class

สำหรับการเตรียมข้อมูลเพื่อจะไปเทรนแบบจำลอง Question Answering นั้น จำเป็นจะต้อง Tokenize ข้อมูลคู่คำถามและบริบท (Question-Context Paired) พร้อม ๆ โดยจะมีผลทดสอบในแต่ละขั้นตอน ดังนี้

1) ทดสอบการตัดข้อความ (Truncation)

จะทำการกำหนดพารามิเตอร์สำหรับการตัดข้อความสำหรับข้อมูลตัวอย่าง ดังนี้ truncation = 'only_second', max_length = 18 และ return_overflowing_tokens =

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TRUE โดยจะแสดงเอาต์พุตก่อนการตัดข้อความ ดังแสดงในรูปที่ 4.2 และเอาต์พุตหลังการตัดข้อความ ซึ่งจะได้ input_ids ออกมา 2 พีเจอร์ ดังแสดงในรูปที่ 4.3

```

1 tokenized_examples = tokenizer(
2   example["question"],
3   example["context"],
4 )
5 tokenized_examples

{'input_ids': [5, 10, 3814, 11851, 68, 483, 6, 6, 10, 3814, 55, 68, 10, 97, 10, 175, 384, 12, 20684, 16821, 6],
'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]}

```

รูปที่ 4.2 เอาต์พุตก่อนทำ Truncation

```

1 tokenized_examples = tokenizer(
2   example["question"],
3   example["context"],
4   truncation="only_second",
5   return_overflowing_tokens=True,
6   max_length=18,
7 )
8 tokenized_examples

{'input_ids': [[5, 10, 3814, 11851, 68, 483, 6, 6, 10, 3814, 55, 68, 10, 97, 10, 175, 384, 6], [5, 10, 3814, 11851, 68, 483, 6, 6, 12, 20684, 16821, 6]], 'attention_mask': [[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]], 'overflow_to_sample_mapping': [0, 0]}

```

รูปที่ 4.3 เอาต์พุตหลังทำ Truncation กรอบสีเขียว คือ พีเจอร์ที่ 1 และกรอบสีเหลือง คือ พีเจอร์ที่ 2

2) ทดสอบการทับซ้อนกันของข้อความ (Overlapping)

จะเพิ่มพารามิเตอร์ stride = 6 หลังจากทำการตัดข้อความ โดยจะแสดงเอาต์พุตก่อนทำ Overlapping ดังแสดงในรูปที่ 4.6 และเอาต์พุตหลังทำ Overlapping ดังแสดงในรูปที่ 4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

1 tokenized_examples = tokenizer(
2   example["question"],
3   example["context"],
4   truncation="only_second",
5   return_overflowing_tokens=True,
6   max_length=18,
7   stride=6,
8 )
9 tokenized_examples

{'input_ids': [[5, 10, 3814, 384, 1500, 6, 6, 10, 3814, 55, 68, 10, 97, 10, 175, 384, 12, 6], [5, 10, 3814, 384, 1500, 6, 6, 10, 97, 10, 175, 384, 12, 20684, 16821, 6]], 'attention_mask': [[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]], 'overflow_to_sample_mapping': [0, 0]}

```

รูปที่ 4.4 เอาต์พุตหลังทำ Overlapping กรอบสี่เหลี่ยม คือ พีเจอร์ที่ 1 และกรอบสี่เหลี่ยม คือ พีเจอร์ที่ 2

3) ทดสอบการทำข้อความให้ยาวเท่ากัน (Padding)

จะเพิ่มพารามิเตอร์ padding="max_length" หลังจากทำ Overlapping โดยจะแสดงเอาต์พุตก่อนทำ Padding ดังแสดงในรูปที่ 4.4 และเอาต์พุตหลังทำ Padding ดังแสดงในรูปที่ 4.5

```

1 tokenized_examples = tokenizer(
2   example["question"],
3   example["context"],
4   truncation="only_second",
5   return_overflowing_tokens=True,
6   max_length=18,
7   stride=6,
8   padding="max_length",
9 )
10 tokenized_examples

{'input_ids': [[5, 10, 3814, 384, 1500, 6, 6, 10, 3814, 55, 68, 10, 97, 10, 175, 384, 12, 6], [5, 10, 3814, 384, 1500, 6, 6, 10, 97, 10, 175, 384, 12, 20684, 16821, 6, 1, 1]], 'attention_mask': [[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0]], 'overflow_to_sample_mapping': [0, 0]}

```

รูปที่ 4.5 เอาต์พุตหลังทำ Padding กรอบสี่เหลี่ยม คือ พีเจอร์ที่ 1 และกรอบสี่เหลี่ยม คือ พีเจอร์ที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และสุดท้ายจะต้องเพิ่มพารามิเตอร์ `return_offsets_mapping = True` เพื่อใช้ในการหาตำแหน่งของช่วงคำตอบโดยจะมีเอาต์พุตรวม ดังแสดงในรูปที่ 4.6

```

1 tokenized_examples = tokenizer(
2   example["question"],
3   example["context"],
4   truncation="only_second",
5   return_overflowing_tokens=True,
6   max_length=18,
7   stride=6,
8   return_offsets_mapping=True,
9   padding="max_length",
10 )
11 tokenized_examples

{'input_ids': [[5, 10, 3814, 384, 1500, 6, 6, 10, 3814, 55, 68, 10, 97, 10, 175, 384, 12, 6], [5, 10, 3814, 384, 1500, 6, 6, 10, 97, 10, 175, 384, 12, 20684, 16821, 6, 1, 1]], 'attention_mask': [[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1], [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0]], 'offset_mapping': [[(0, 0), (0, 1), (0, 4), (4, 9), (9, 15), (0, 0), (0, 0), (0, 1), (0, 4), (4, 8), (8, 10), (10, 11), (11, 12), (12, 13), (13, 18), (18, 23), (23, 26), (0, 0)], [(0, 0), (0, 1), (0, 4), (4, 9), (9, 15), (0, 0), (0, 0), (10, 11), (11, 12), (12, 13), (13, 18), (18, 23), (23, 26), (26, 31), (31, 40), (0, 0), (0, 0), (0, 0)]], 'overflow_to_sample_mapping': [0, 0]}

```

รูปที่ 4.6 เอาต์พุตหลังจากการทำ Tokenization

เมื่อสังเกตในรูปที่ 4.6 ที่ `'overflow_to_sample_mapping'` จะพบว่ามีค่า คือ `[0, 0]` ซึ่งหมายความว่า `input_ids` ของเอาต์พุตมี 2 พีเจอร์ เกิดจากการที่คู่คำถามและบริบทจากข้อมูลตัวอย่างถูกทำ Truncation ทำให้เกิดพีเจอร์ขึ้นมาใหม่อีกหนึ่งพีเจอร์ โดยจะแยกพีเจอร์ทั้งสองของเอาต์พุต มาแสดงในรูปที่ 4.7 และรูปที่ 4.8 ตามลำดับ

```

input_ids [5, 10, 3814, 384, 1500, 6, 6, 10, 3814, 55, 68, 10, 97, 10, 175, 384, 12, 6]
attention_mask [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
offset_mapping [(0, 0), (0, 1), (0, 4), (4, 9), (9, 15), (0, 0), (0, 0), (0, 1), (0, 4), (4, 8), (8, 10), (10, 11), (11, 12), (12, 13), (13, 18), (18, 23), (23, 26), (0, 0)]
overflow_to_sample_mapping 0

```

รูปที่ 4.7 เอาต์พุตพีเจอร์ที่ 1 ของคู่คำถามและบริบท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่” และบริบทของพีเจอร์ที่ 2 คือ “ 4 กำลังเรียนที่วิเศษลาดกระบัง” ซึ่งมีข้อความที่ซ้อนทับกันคือ “4 กำลังเรียนที่”

จาก attention_mask ของทั้งสองพีเจอร์ จะพบว่าพีเจอร์ที่ 1 จะมีค่าเป็น 1 ทั้งหมด แต่ในพีเจอร์ที่ 2 จะมี 0 อยู่ 2 ค่า นั่นก็คือโทเคน <pad> นั่นเอง เพราะ Self-attention Mechanism จะไม่สนใจโทเคน <pad>

และสุดท้าย offset_mapping คือ Tuple ที่ใช้บอกว่าแต่ละโทเคนมีความยาวของ Character เท่าใด เช่น โทเคนของคำว่า “พลอย” จะมี Offset_mapping คือ (0, 4) ซึ่งจะมีความยาว Character เท่ากับ 4 และในโทเคนถัดจาก “พลอย” คือ “เรียน” จะมี Offset_mapping คือ (4, 9) ซึ่งจะมีความยาว Character เท่ากับ 5 แต่สมาชิกใน Tuple จะมีการนับเลขต่อจาก Tuple ก่อนหน้า

จากเอาต์พุตของกระบวนการ Tokenization สำหรับข้อมูลด้วย AutoTokenizer Class ดังรูปที่ 4.6 สามารถ Tokenize ชุดข้อมูลตัวอย่างได้อย่างถูกต้อง พร้อมนำไปทำกระบวนการหา “start_position” และ “end_position” ต่อไป

4.1.1.2 ทดสอบการหาตำแหน่งของช่วงคำตอบ (Answer Span)

ในหัวข้อนี้จะทำการทดสอบว่าโทเคน “start_positions” และ “end_positions” โดยทดสอบกับชุดข้อมูลตัวอย่าง

ค่าของโทเคน “start_positions” และ “end_positions” โดยพีเจอร์ที่ 1 มีค่าของ “start_positions” และ “end_positions” เท่ากับ 0 หรือหมายความว่าไม่มีช่วงคำตอบ อยู่ในพีเจอร์ที่ 1

ส่วนพีเจอร์ที่ 2 จะมีค่าของ “start_positions” เท่ากับ 11 และมีค่าของ “end_positions” เท่ากับ 15 หรือหมายความว่ามีความยาวช่วงคำตอบ (Answer Span) อยู่ในพีเจอร์ที่ 2 ณ ตำแหน่งที่ 11 ถึง 15 ดังแสดงในรูปที่ 4.11

Feature	start_positions	end_positions
1	0	0
2	11	15

รูปที่ 4.11 ค่าของ “start_positions” และ “end_positions” ทั้ง 2 Feature จากชุดข้อมูล ตัวอย่างแสดงตำแหน่งของช่วงคำตอบ (Answer Span) ในบริบท (Context)

คำตอบที่ถูกถอดรหัส (Decode) จากพีเจอร์ที่ 1 และ 2 ของ “input_ids” (Labels Give Answer) เปรียบเทียบกับคำตอบจริง (Theoretical Answer) พบว่ากระบวนการหาช่วงคำตอบจากบริบทนั้นมีความถูกต้อง ดังแสดงในรูปที่ 4.12 และ 4.13

Labels Give Answer	Theoretical Answer
เรียนที่วิศวะลาดกระบัง	เรียนที่วิศวะลาดกระบัง

รูปที่ 4.12 คำตอบที่ได้จาก “start_position” และ “end_position” (Labels Give Answer) เปรียบเทียบกับคำตอบจริงจากชุดข้อมูลตัวอย่าง (Theoretical Answer) ของพีเจอร์ที่ 1

Labels Give Answer	Theoretical Answer
เรียนที่วิศวะลาดกระบัง	เรียนที่วิศวะลาดกระบัง

รูปที่ 4.13 คำตอบที่ได้จาก “start_position” และ “end_position” (Labels Give Answer) เปรียบเทียบกับคำตอบจริงจากชุดข้อมูลตัวอย่าง (Theoretical Answer) ของพีเจอร์ที่ 2

จากการทดสอบกระบวนการเตรียมข้อมูลนั้น จะมีกระบวนการที่สำคัญคือ Tokenization และกระบวนการหาตำแหน่งของช่วงคำตอบ (Answer Span) เพื่อให้ได้เอาต์พุตที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พร้อมจะนำไปเทรน wangchanBERTa Pre-trained Model หรือก็คือกระบวนการทำ Fine-tuning

4.1.2 ทดสอบกระบวนการหาพารามิเตอร์ที่ดีที่สุด (Hyperparameter Optimization)

ผลลัพธ์ของพารามิเตอร์ที่ได้จากกระบวนการ Hyperparameter Optimization แสดงดังรูปที่ 4.14

Learning Rate	Weight Decay	Epoch
8.029540331783353e-05	0.00013063434329338937	3

รูปที่ 4.14 Hyperparameter จาก Hyperparameter Optimization

4.1.3 กระบวนการประเมินประสิทธิภาพของแบบจำลองด้วยผลคะแนน Exact Match และ F1

ผลคะแนน Exact Match และ F1 จากการประเมินประสิทธิภาพแบบจำลอง แสดงดังรูปที่ 4.15

Exact Match	F1
52.380952	75.741713

รูปที่ 4.15 ผลคะแนน Exact Match และ F1

4.2 ทดสอบการใช้งาน Knowledge Base

4.2.1 ทดสอบการเรียกดูข้อมูลบริบทจากตารางสำหรับเก็บข้อมูลบริบท

ตารางสำหรับเก็บข้อมูลบริบทถูกสร้างขึ้นเพื่อเก็บข้อมูลบริบทหรือคลังความรู้ทั้งหมดไว้ ข้อมูลบริบทต่าง ๆ ที่ถูกเก็บไว้ในตารางนี้จะถูก Retriever ดึงไปใช้ ดังนั้น การทดสอบนี้จะทำการเรียกดูข้อมูลบริบทจาก Knowledge Base ตารางสำหรับเก็บข้อมูลบริบทมาแสดง ด้วยคำสั่ง SQL SELECT ดังแสดงในรูปที่ 4.16

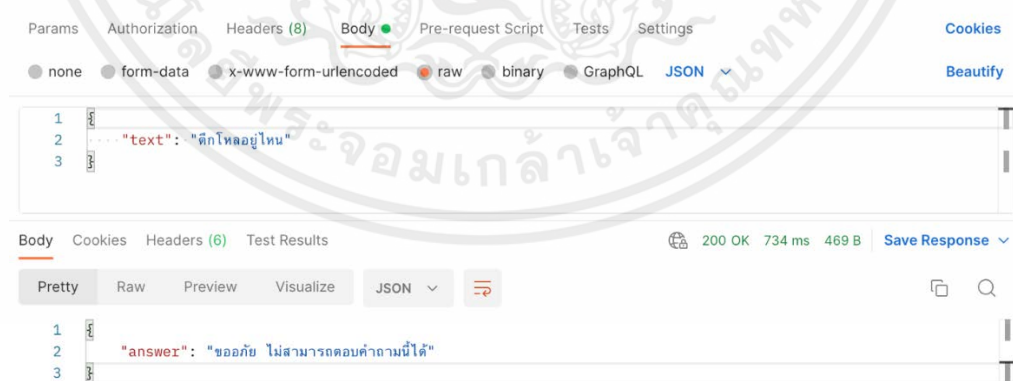
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

id	context	created_at
1	ชื่อหลักสูตรภาษาไทย คือ หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรม โทรคมนาคมและโค...	2023-03-22 06:28:40
2	ชื่อหลักสูตรภาษาอังกฤษ คือ Bachelor of Engineering in Telecommunications and Network E...	2023-03-22 06:28:40
3	ชื่อเต็มภาษาไทยปริญญาและสาขาวิชา คือ วิศวกรรมศาสตรบัณฑิต (วิศวกรรม โทรคมนาคมและเคร...	2023-03-22 06:28:40
4	ชื่อย่อภาษาไทยปริญญาและสาขาวิชา คือ ว.บ. (วิศวกรรม โทรคมนาคมและ โครงข่าย)	2023-03-22 06:28:40
5	ชื่อเต็มภาษาอังกฤษปริญญาและสาขาวิชา คือ Bachelor of Engineering (Telecommunications a...	2023-03-22 06:28:40

รูปที่ 4.16 ผลการทดสอบการเรียกดูข้อมูลบริบทตัวอย่าง จาก Knowledge Base ตารางสำหรับเก็บข้อมูลบริบท

4.2.2 ทดสอบการเพิ่มคำถามไปที่ตารางสำหรับคำถามที่ระบบไม่สามารถตอบได้

ดังที่ได้อธิบายไว้ในข้อที่ 3.1.3.2 ตารางสำหรับเก็บข้อมูลคำถามที่ระบบไม่สามารถตอบได้ ถูกสร้างขึ้นเพื่อเก็บข้อมูลคำถามจากผู้ใช้งานที่ซึ่งระบบไม่สามารถตอบได้ เพื่อนำคำถามดังกล่าวมาพัฒนาและปรับปรุงระบบ ดังนั้นการทดสอบนี้ จะทำการทดสอบเพื่อดูคำถามที่ระบบไม่สามารถตอบได้ ซึ่งจะถูเก็บหรือบันทึกไว้ในตาราง แสดงดังรูปที่ 4.17 จากรูป ผู้จัดทำได้ทำการทดสอบระบบด้วยคำถาม ‘ตึกโหลอยู่ไหน’ และระบบไม่สามารถตอบคำถามนี้ได้ เนื่องจากคำถามอยู่นอกขอบเขตที่สามารถตอบได้ และเนื่องจากระบบไม่สามารถตอบคำถามนี้ได้ คำถามดังกล่าวจะถูกเพิ่มไปยัง Knowledge Base ที่ตารางสำหรับคำถามที่ระบบไม่สามารถตอบได้ แสดงดังรูปที่ 4.18 ID ที่ 198



รูปที่ 4.17 ผลการทดสอบระบบด้วยคำถามที่ระบบไม่สามารถตอบได้ ผ่าน Postman

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

id	question	inserted_at
197	โรงเอออยู่ไหน	2023-03-25 11:20:47
198	ตึก ไทลอยู่ไหน	2023-03-25 11:21:01

รูปที่ 4.18 คำถามที่ระบบไม่สามารถตอบได้ ที่ถูกเพิ่มไปยัง Knowledge Base ตารางสำหรับ คำถามที่ระบบไม่สามารถตอบได้

4.3 ทดสอบระบบ Recommendation System API

สำหรับการทดสอบการทำงานของระบบ ผู้จัดทำได้จัดแบ่งผลการทดสอบออกเป็นระบบย่อยดังต่อไปนี้

4.3.1 ทดสอบระบบตรวจสอบการตกทาย

ผลการทดสอบระบบตรวจสอบการตกทาย แสดงดังรูปที่ 4.19 โดยผู้จัดทำได้ทำการป้อนข้อความ 'สวัสดีค่ะ' ให้กับระบบ และระบบได้โต้ตอบด้วยข้อความ 'สวัสดี ยินดีให้บริการ' เป็นไปตามดังที่ได้ออกแบบไว้

none form-data x-www-form-urlencoded raw binary GraphQL JSON

```

1 {
2   "text": "สวัสดีค่ะ"
3 }

```

Body Cookies Headers (6) Test Results 200 OK 330 ms 449 B

Pretty Raw Preview Visualize JSON

```

1 {
2   "answer": "สวัสดีครับ ยินดีให้บริการ"
3 }

```

รูปที่ 4.19 ผลการทดสอบระบบตรวจสอบการตกทาย ผ่าน Postman

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3.2 ทดสอบระบบเตรียมข้อมูลคำถาม (Question Preprocessing)

สำหรับผลการทดสอบระบบเตรียมข้อมูล ซึ่งระบบทำการแบ่งข้อความออกเป็นคำย่อย ๆ แสดงดังรูปที่ 4.20 โดยคำถาม ‘ชื่อหลักสูตรภาษาไทยคืออะไร’ จะเป็นคำถามที่ผู้จัดทำจะใช้ในการทดสอบในหัวข้อที่ 4.3.2 ถึง 4.3.5

Raw Question: ชื่อหลักสูตรภาษาไทยคืออะไร
Preprocessed Question: ['ชื่อ', 'หลักสูตร', 'ภาษาไทย', 'คืออะไร']

รูปที่ 4.20 ผลการทดสอบระบบเตรียมข้อมูลคำถาม

4.3.3 ทดสอบระบบเตรียมข้อมูลบริบท (Contexts Preprocessing)

ผลการทดสอบระบบเตรียมข้อมูลบริบท แสดงดังรูปที่ 4.21

Raw Context : ชื่อหลักสูตรภาษาไทย คือ หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคมและโครงข่าย
Preprocessed Context : ['ชื่อ', 'หลักสูตร', 'ภาษาไทย', 'คือ', 'หลักสูตร', 'วิศวกรรม', 'ศาสตร์', 'บัณฑิต', 'สาขาวิชา', 'วิศวกรรม', 'โทรคมนาคม', 'และ', 'โครงข่าย']

รูปที่ 4.21 ตัวอย่างบริบทก่อน-หลังผ่านระบบเตรียมข้อมูลบริบท

4.3.4 ทดสอบระบบค้นคืนข้อมูล (Retriever)

ดังที่ได้อธิบายไว้ในหัวข้อที่ 3.1.4.4 ระบบสืบค้นคืนข้อมูล หรือ Retriever เป็นระบบที่ทำหน้าที่ค้นหาบริบท (Context) ที่เกี่ยวข้อง จากคำถามของผู้ใช้งาน โดยจะใช้อัลกอริทึม Okapi BM25 ซึ่งจะมีวิธีการคือเปรียบเทียบความถี่ของคำจากคำถามกับบริบททั้งหมดที่ถูกเก็บไว้ใน Knowledge Base แล้วส่งคืนบริบทที่เกี่ยวข้องกับคำถามมากที่สุดกลับมา ดังนั้นการทดสอบนี้ จะเป็นการทดสอบหาบริบทที่เกี่ยวข้องกับคำถาม ‘ชื่อหลักสูตรภาษาไทยคืออะไร’ มากที่สุด 2 อันดับแรก ที่ได้จากระบบค้นคืนข้อมูล ผลการทดสอบแสดงดังรูปที่ 4.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

(Score: 15.9761) ชื่อหลักสูตรภาษาไทย คือ หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคมและโครงข่าย
(Score: 11.4261) ชื่อเต็มภาษาไทยปริญญาและสาขาวิชา คือ วิศวกรรมศาสตรบัณฑิต (วิศวกรรมโทรคมนาคมและโครงข่าย)

รูปที่ 4.22 ผลการทดสอบระบบค้นคืนข้อมูล เพื่อหาบริบทที่เกี่ยวข้องกับคำถามมากที่สุด

4.3.5 ทดสอบการใช้งาน Reader Model

เมื่อได้นำคำถามของผู้ใช้งานไปเข้าระบบสืบค้นคืนข้อมูล จะได้บริบทที่เกี่ยวข้องจาก Knowledge Base มาในหัวข้อที่ 4.2.4 เมื่อนำคำถามตัวอย่างและบริบทนี้ป้อนเข้าสู่ Reader Model จะได้คำตอบออกมาดังแสดงในรูปที่ 4.23 โดยมีคะแนนอยู่ที่ 0.9256 คะแนน

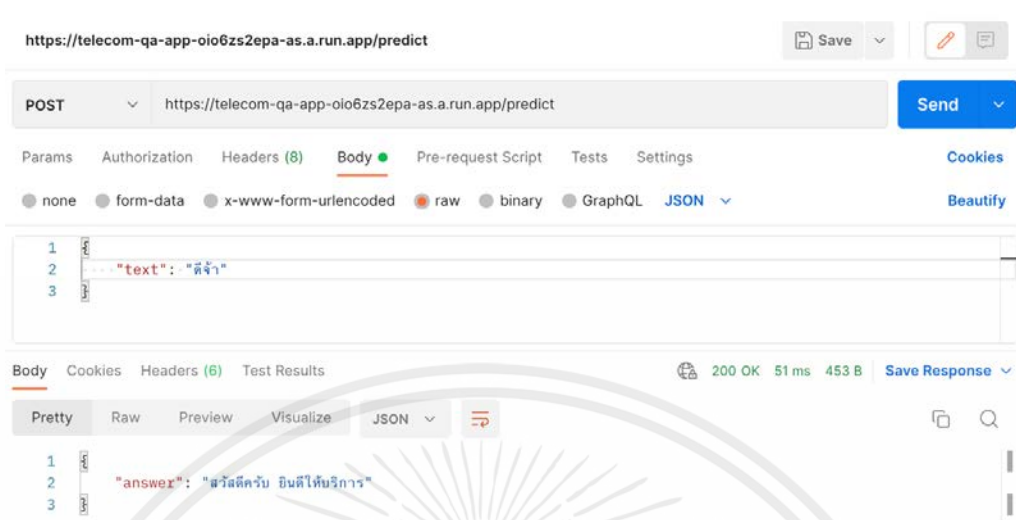
```
{'score': 0.9256300330162048,
  'start': 0,
  'end': 89,
  'answer': 'ชื่อหลักสูตรภาษาไทย คือ หลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมโทรคมนาคมและโครงข่าย'}
```

รูปที่ 4.23 ผลการทดสอบ Reader Model

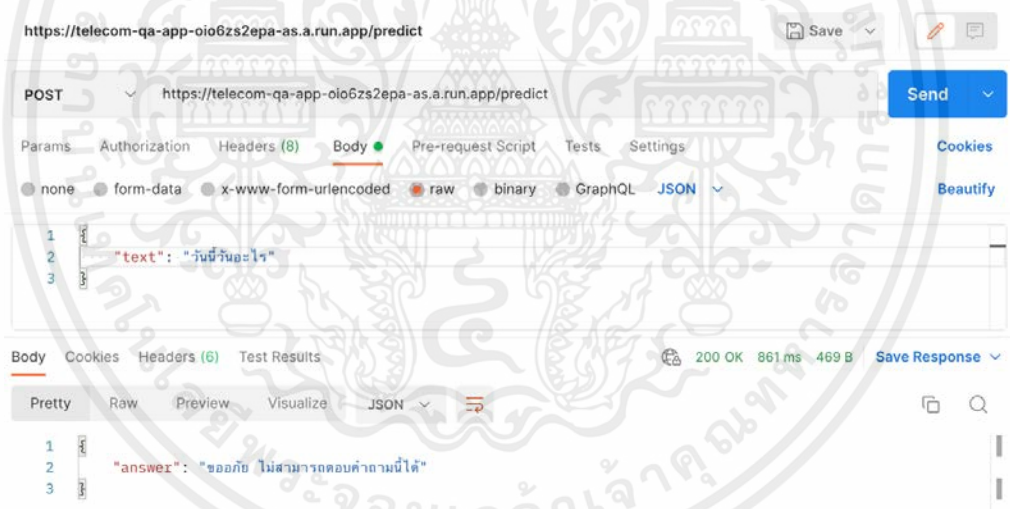
4.3.6 ทดสอบการเชื่อมต่อ Recommendation System API ด้วยโปรแกรม Postman

เมื่อได้ระบบตามที่ต้องการ และได้ทำการ Deploy ตัวระบบเป็นที่เรียบร้อยแล้ว การทดสอบนี้ ผู้จัดทำจะทำการทดสอบการตอบคำถาม รวมถึงการโต้ตอบอื่น ๆ ของระบบโดยรวมผ่าน API ที่ผู้จัดทำได้สร้างขึ้น ซึ่งได้ทำการทดสอบผ่าน Postman โดยผลการทดสอบ API ด้วยข้อความต่าง ๆ แสดงดังรูปที่ 4.24, 4.25 และ 4.26

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

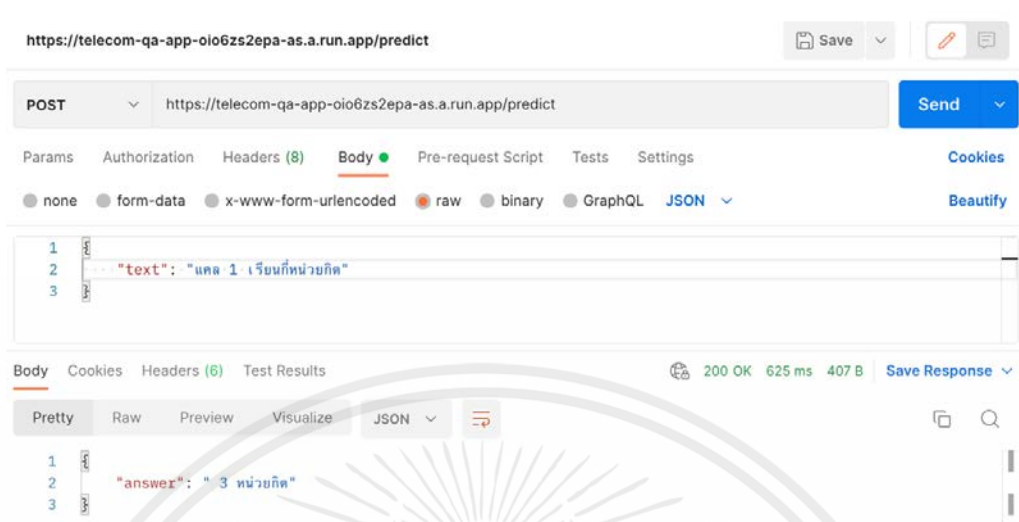


รูปที่ 4.24 การทดสอบ API ของระบบโดยรวม ผ่าน Postman โดยทดสอบด้วยข้อความทักทาย



รูปที่ 4.25 การทดสอบ API ของระบบโดยรวม ผ่าน Postman โดยทดสอบด้วยคำถามนอกขอบเขตการตอบคำถามของระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

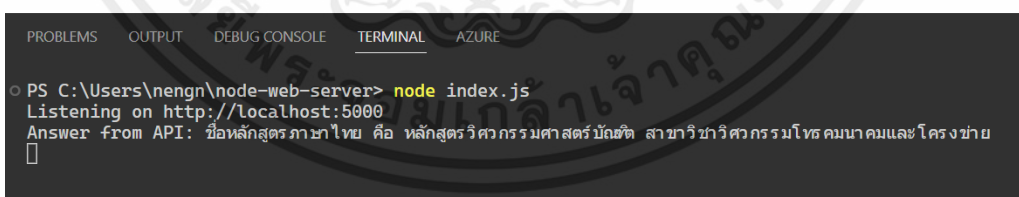


รูปที่ 4.26 การทดสอบ API ของระบบโดยรวม ผ่าน Postman โดยทดสอบด้วยคำถามที่เกี่ยวข้อง
หลักสูตรภาควิชาโทรคมนาคม

4.4 ทดสอบการสื่อสารระหว่าง Web Server กับ API

4.4.1 ทดสอบการส่ง-รับข้อมูลที่ Web Server กับ Recommendation System API

การทดสอบการร้องขอข้อมูลจาก Web Server ไปยัง Recommendation System API ด้วยคำสั่ง `fetch()` ดังหัวข้อที่ 3.1.8.3 มีผลทดสอบดังแสดงในรูปที่ 4.27

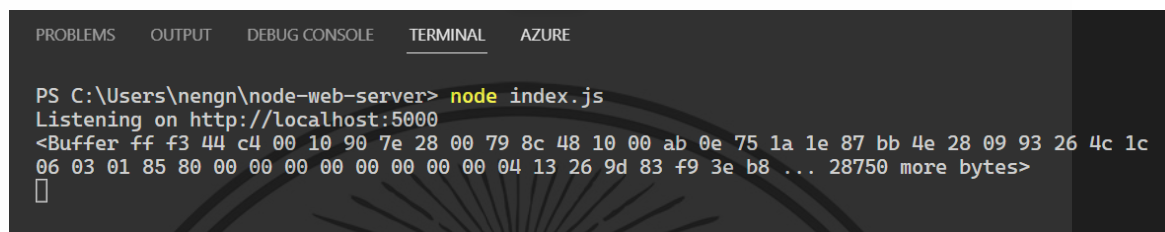


รูปที่ 4.27 การตอบสนอง (Response) จาก Recommendation System API ที่ Terminal ของ
เว็บเซิร์ฟเวอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4.2 ทดสอบการส่ง-รับข้อมูลที่เว็บเซิร์ฟเวอร์กับ Google Cloud Text-to-speech API

การทดสอบการร้องขอข้อมูลจาก Web Server ไปยัง Google Cloud Text-to-speech API ดังหัวข้อที่ 3.1.8.4 โดยข้อมูลที่ได้กลับมาจะอยู่ในรูปแบบ Buffer ดังแสดงในรูปที่ 4.28



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE
PS C:\Users\nengn\node-web-server> node index.js
Listening on http://localhost:5000
<Buffer ff f3 44 c4 00 10 90 7e 28 00 79 8c 48 10 00 ab 0e 75 1a 1e 87 bb 4e 28 09 93 26 4c 1c
06 03 01 85 80 00 00 00 00 00 00 04 13 26 9d 83 f9 3e b8 ... 28750 more bytes>

```

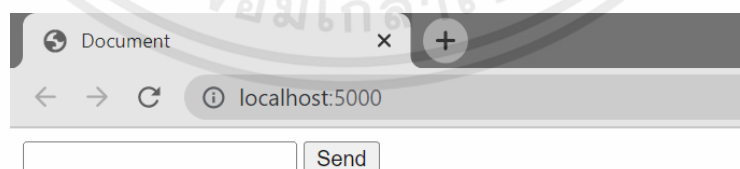
รูปที่ 4.28 การตอบสนอง (Response) จาก Google Cloud Text-to-speech API ที่ Terminal ของ Web Server

4.5 ทดสอบการสื่อสารระหว่างฝั่งผู้ใช้งานกับเว็บเซิร์ฟเวอร์

การทดสอบการสื่อสารระหว่างฝั่งผู้ใช้งาน (Client) กับ Web Server) จะเริ่มต้นจากเว็บเซอร์ผู้ใช้ส่งคำขอหน้าเว็บไซต์กับ Web Server และหลังจากนั้นคือการสื่อสารกัน

4.5.1 ทดสอบการร้องขอของหน้าเว็บไซต์ด้วยเว็บเบราว์เซอร์

จากหัวข้อที่ 3.1.9 เมื่อผู้ใช้ทำการร้องขอ (Request) แบบเมธอด GET มาที่ URL <http://localhost:5000/> Web Server จะตอบสนองกลับด้วยการส่งหน้าเว็บไซต์ที่ได้ออกแบบให้แก่ผู้ใช้ ดังแสดงในรูปที่ 4.29



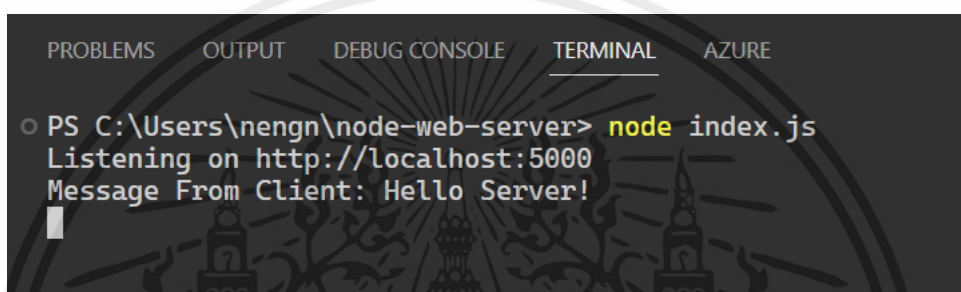
รูปที่ 4.29 หน้าเว็บไซต์ที่เว็บเซิร์ฟเวอร์ส่งกลับให้ผู้ใช้ (Client)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5.2 ทดสอบการส่ง-รับข้อมูลระหว่างฝั่งผู้ใช้งานกับเว็บเซิร์ฟเวอร์

จากหัวข้อที่ 4.5.1 ทำการพิมพ์ข้อความ “Hello Server!” แล้วกดปุ่ม Send เพื่อส่งข้อความไปยัง Web Server เมื่อฝั่ง Server ได้รับข้อความดังแสดงในรูปที่ 4.30 ก็จะมีการตอบกลับว่า “Hello, Client!!!” ดังแสดงในรูปที่ 4.31

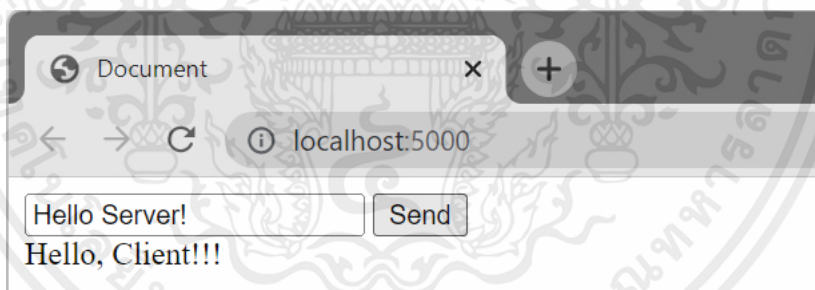
ดังนั้นจึงถือว่าการสื่อสารระหว่างฝั่งผู้ใช้งานกับ Web Server สามารถสื่อสารกันได้



```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL AZURE
PS C:\Users\nengn\node-web-server> node index.js
Listening on http://localhost:5000
Message From Client: Hello Server!
  
```

รูปที่ 4.30 Terminal ของเว็บเซิร์ฟเวอร์เมื่อได้รับข้อความจากผู้ใช้



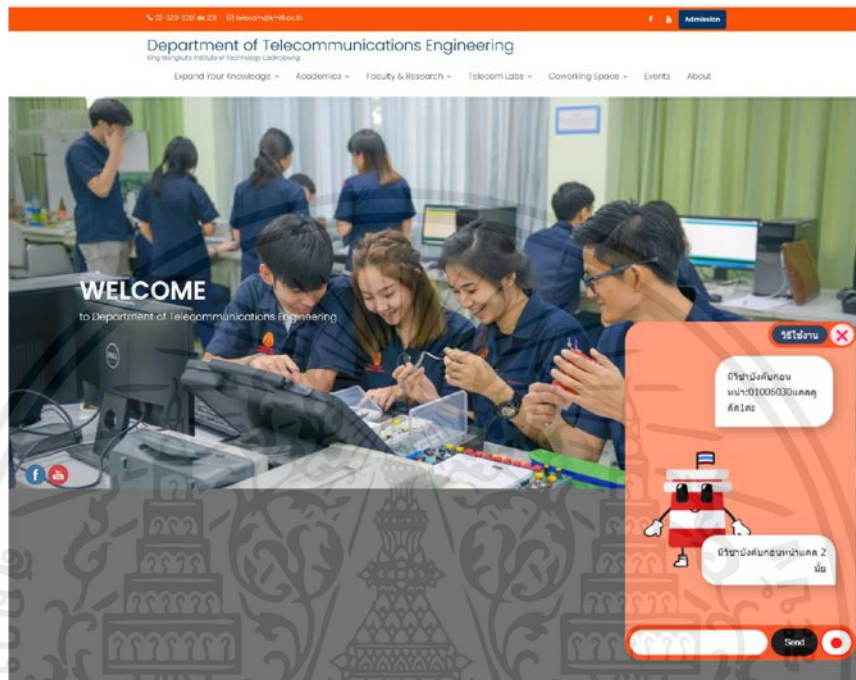
รูปที่ 4.31 หน้าเว็บไซต์หลังจากที่เว็บเซิร์ฟเวอร์ตอบกลับ

4.6 ทดสอบการใช้งานเว็บไซต์ที่ขนาดหน้าจอต่างกัน (Responsive Web Design)

สำหรับการออกแบบหน้าเว็บไซต์ให้รองรับหน้าจอขนาดที่ต่างกัน จะรองรับการใช้งานในอุปกรณ์ Desktop ที่ขนาดความกว้างอย่างต่ำ 1366 px ดังแสดงในรูปที่ 4.32 และหากมีค่า

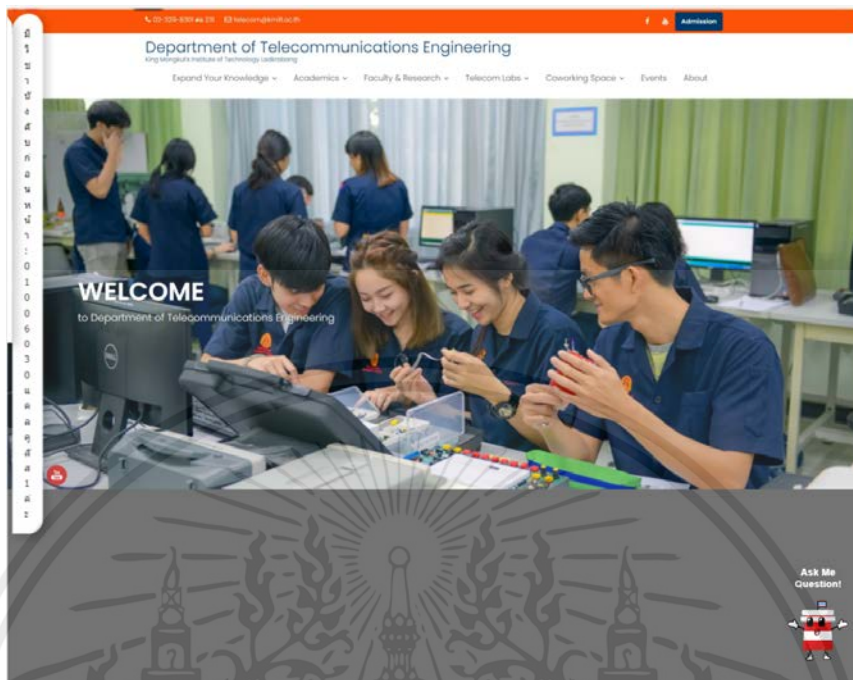
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

น้อยกว่านั้นจะไม่สามารถใช้งานได้ดังแสดงในรูปที่ 4.33 และจะใช้งานได้ที่ดีที่สุดคือขนาด 1920x1080 px ดังแสดงในรูปที่ 4.34

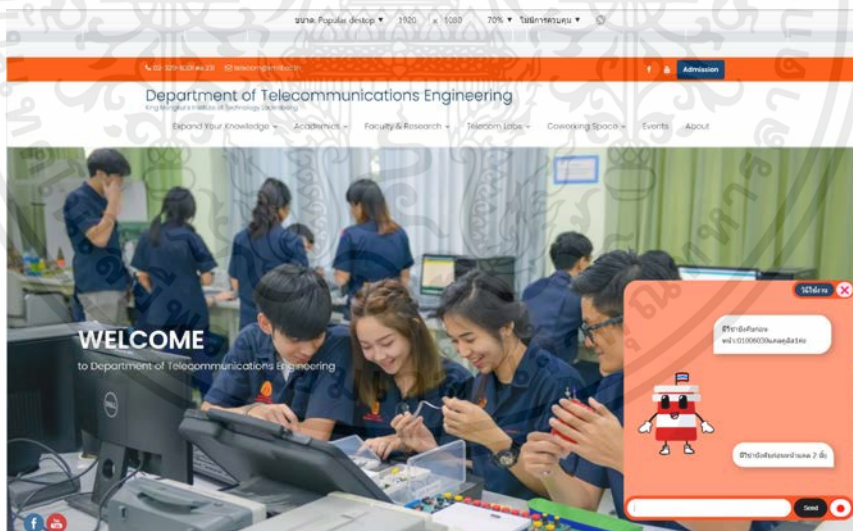


รูปที่ 4.32 หน้าเว็บไซต์ขนาดความกว้างขั้นต่ำที่รองรับหรือขนาดความกว้าง 1366px

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.33 หน้าเว็บไซต์ขนาดความกว้างที่ไม่รองรับหรือขนาดความกว้างน้อยกว่า 1366px

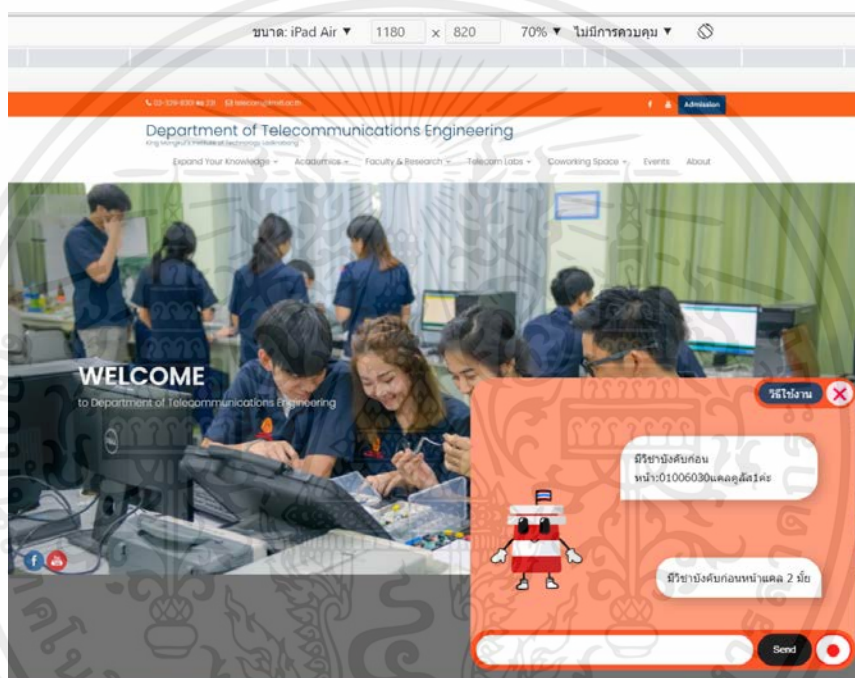


รูปที่ 4.34 หน้าเว็บไซต์ที่สามารถใช้งานได้ดีที่สุดที่ขนาด 1920x1080 px

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

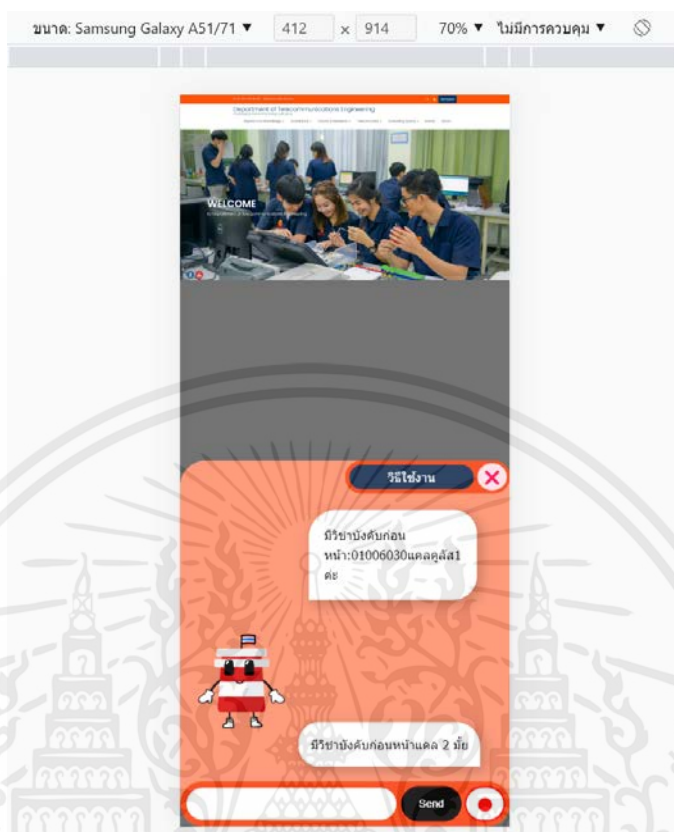
นอกจากอุปกรณ์เดสก์ท็อปแล้วยังสามารถใช้งานในแท็บเล็ตได้โดยต้องมีขนาดความกว้างอยู่ระหว่าง 769px - 1365px โดยจะยกตัวอย่าง iPad air ซึ่งมีหน้าจอขนาด 1180x820px ดังแสดงในรูปที่ 4.35

และสุดท้ายจะรองรับการใช้งานในโทรศัพท์มือถือที่มีขนาดหน้าจออยู่ระหว่าง 320px - 768px โดยจะยกตัวอย่าง Samsung Galaxy A51/71 ซึ่งมีหน้าจอขนาด 412x914px ดังแสดงในรูปที่ 4.36



รูปที่ 4.35 หน้าเว็บไซต์ที่ขนาด 1180x820px (iPad air)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

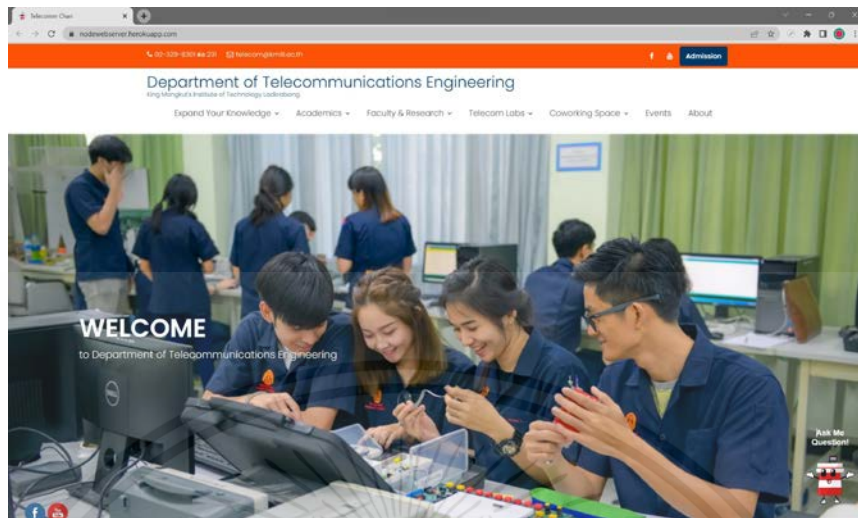


รูปที่ 4.36 หน้าเว็บไซต์ที่ขนาด 412x914px (Samsung Galaxy A51/71)

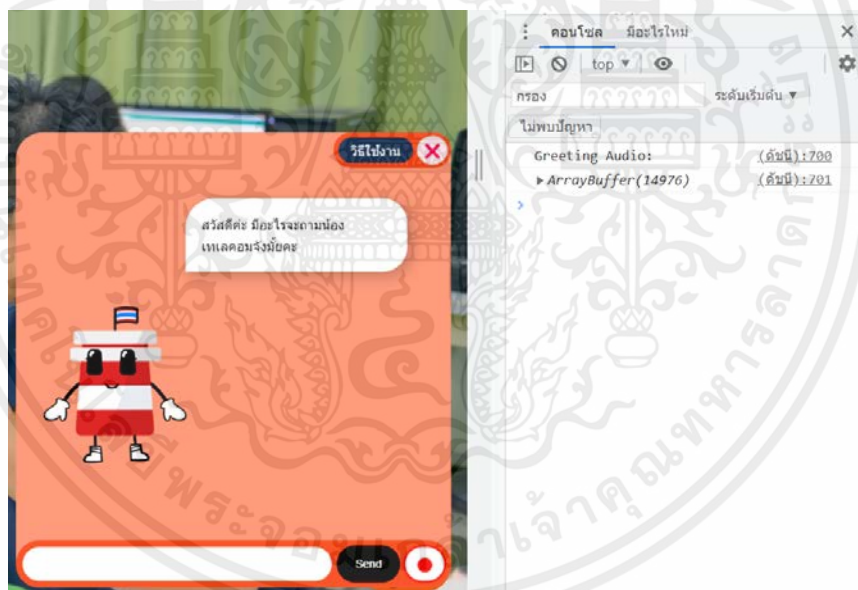
4.7 ทดสอบการใช้งานระบบรวม

ในขั้นตอนนี้ผู้จัดทำได้ทำการทดสอบการใช้งานระบบรวม โดยเริ่มต้นจากเข้าไปที่หน้าเว็บไซต์ <https://nodewebserver.herokuapp.com/> เพื่อร้องขอหน้าเว็บไซต์จาก Web Server ดังแสดงในรูปที่ 4.37 จากนั้นกดปุ่ม Ask Me Question! เพื่อเข้าใช้งาน Chat-popup โดยจะมีข้อความต้อนรับคือ “สวัสดีค่ะ มีอะไรจะถามน้องเทเลคอมจิ้งมัยคะ” พร้อมเสียงพูดจากการ์ตูนกราฟิกเคลื่อนไหว (Motion Graphic) โดยสามารถสังเกตว่าระบบสามารถใช้งาน Text-to-speech ได้จากการดูข้อมูลใน Console ของเว็บเบราว์เซอร์ด้วยการกด ctrl+shift+I โดยจะมีข้อความกำกับว่า “Greeting Audio” สำหรับข้อความเสียงนี้ดังแสดงในรูปที่ 4.38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



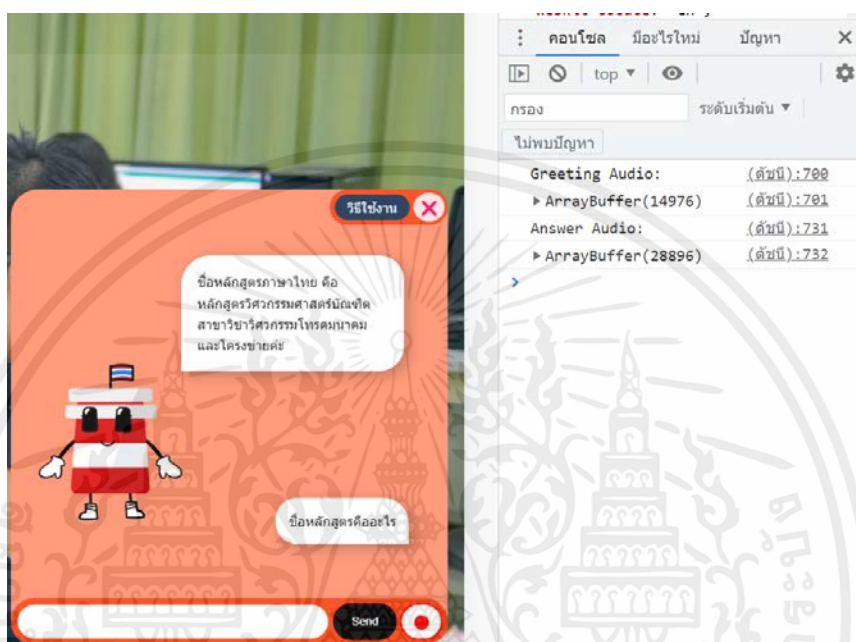
รูปที่ 4.37 หน้าเว็บไซต์ของ nodewebserver.herokuapp.com



รูปที่ 4.38 ผลลัพธ์เมื่อกดปุ่ม Ask Me Question! และหน้า Console บนเว็บเบราว์เซอร์แสดงผลลัพธ์ข้อมูลไฟล์เสียง Greeting Audio

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากนั้นทดลองถามคำถาม “ชื่อหลักสูตรคืออะไร” โดยการพิมพ์ลงในช่องกรอกข้อความ จะได้ข้อความคำตอบกลับมาพร้อมเสียงพูด โดยข้อมูลเสียงนี้จะมีข้อความกำกับไว้ว่า “Answer Audio” ดังแสดงในรูปที่ 4.39



รูปที่ 4.39 ผลลัพธ์เมื่อถามคำถามผ่านการพิมพ์ในช่องกรอกข้อความและหน้า Console บน Web Browser แสดงผลลัพธ์ข้อมูลไฟล์เสียง Answer Audio

ต่อมาจะทดลองใช้งานถามคำถามผ่านเสียงพูด หรือ Speech-to-text โดยการกดปุ่มบันทึกเสียง ซึ่งในขณะที่กำลังบันทึกเสียงนั้นข้อความที่พูดจะขึ้นมาแสดงที่หน้าต่างนี้และปุ่มบันทึกเสียงจะเปลี่ยนเป็นรูปสี่เหลี่ยมสีแดง ดังแสดงในรูปที่ 4.40 และเมื่อต้องการส่งคำถามนี้ให้ทำการกดปุ่มบันทึกเสียงอีกครั้งเพื่อเป็นการหยุดบันทึกและส่งคำถามนี้ไปสู่ระบบหลังบ้าน (Backend) โดยข้อความคำตอบที่ระบบหลังบ้านตอบกลับนั้นคือ “ขอภัย ไม่สามารถตอบคำถามนี้ได้ค่ะ” ดังแสดงในรูปที่ 4.41



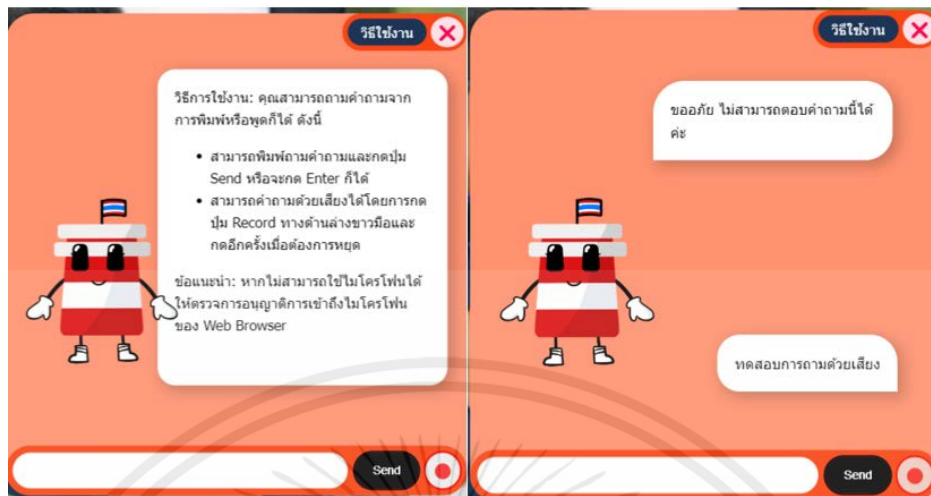
รูปที่ 4.40 หน้าต่าง Chat-popup ในขณะที่กำลังบันทึกเสียง



รูปที่ 4.41 ผลลัพธ์เมื่อถามคำถามผ่านการใช้งาน speech-to-text

และสุดท้ายจะทำการทดสอบกดปุ่ม “วิธีใช้” ซึ่งเมื่อทำการกดแล้วจะแสดงข้อความ วิธีใช้ออกมา และเมื่อทำการกดปุ่มอีกครั้งหน้าต่าง Chat-popup จะกลับสู่สถานะเดิมก่อนการกดปุ่ม ดังแสดงในรูปที่ 4.42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.42 ผลลัพธ์เมื่อทำการกดปุ่มวิธีใช้ (ซ้าย) และกลับสู่สถานะเดิมเมื่อกดปุ่มอีกครั้ง (ขวา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปฏิญานิพนธ์นี้นำเสนอการศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ปฏิญานิพนธ์นี้เป็นการออกแบบระบบเป็น 4 ส่วน ได้แก่ ส่วนที่หนึ่งคือ Web Browser ที่ผู้ใช้งานสามารถถามคำถามผ่านการพิมพ์ข้อความหรือเสียงพูดผ่านเว็บไซต์ ส่วนที่สองคือ Web Server สำหรับรับคำขอและส่งคืนคำขอของผู้ใช้งานผ่านทาง Web Browser ส่วนที่สามคือ Recommendation System API สำหรับเปิดให้บริการกับนักพัฒนาเว็บแอปพลิเคชัน หรือหน่วยงานต่าง ๆ ที่ต้องการใช้ระบบถาม - ตอบ ในข้อมูลของตนเอง ส่วนสุดท้ายคือ Google Cloud Text-to-Speech API ใช้สำหรับกระบวนการ Text-to-Speech ซึ่งในส่วนแรกได้ทำการออกแบบและสร้างเพื่อให้ผู้ใช้งานเข้ามาใช้งานเพื่อถามคำถามเกี่ยวกับวิศวกรรมโทรคมนาคม สจล. โดยผ่านทางหน้าต่าง Chat-Popup ผ่านการพิมพ์หรือสามารถกดปุ่มบันทึกเสียงเพื่อถามคำถามผ่านเสียงพูด ซึ่งคำถามที่ผู้ใช้งานสามารถถามได้นั้นต้องจัดอยู่ในคำถามประเภทของความจำ เช่น ชื่อหลักสูตรคืออะไร, แคลคูลัส 2 เรียนกี่หน่วยกิต, จำนวนหน่วยกิตตลอดปีการศึกษา เนื่องมาจากการจัดทำชุดข้อมูลของปฏิญานิพนธ์นี้อ้างอิงบริบทและคำถามจากเล่มหลักสูตรภาควิชาวิศวกรรมโทรคมนาคม สจล. แล้วจึงนำชุดข้อมูลเหล่านี้ไปให้แบบจำลองเรียนรู้ แบบจำลองจึงสามารถตอบคำถามโดยการสกัดเนื้อความจากบริบทที่เรียนรู้มาเท่านั้น ดังนั้นคำถามที่ผู้ใช้งานไม่ควรถามจึงเป็นคำถามประเภทที่ต้องใช้การตัดสินใจหรือต้องแนะนำซึ่งคำถามเหล่านี้อยู่นอกเหนือเล่มหลักสูตรภาควิชาวิศวกรรมโทรคมนาคม สจล. เช่น หลักสูตรไหนเหมาะกับฉัน หรือลงเรียนวิชาเลือกอะไรดี ส่วนที่สองสร้างขึ้นเพื่อรับคำร้องขอจากผู้ใช้งานส่งต่อไปให้ Recommendation System API ตอบคำถามแล้วส่งคำถามกลับมา จากนั้นคำตอบนี้จะถูกส่งไปให้ผู้ใช้งานพร้อมกับ Google Cloud Text-to-Speech API ส่วนที่สามเป็นส่วนที่สร้างขึ้นมาเพื่อเก็บระบบ Pre-processing, Retriever, Knowledge Base และ Model และส่วนสุดท้ายสร้างเพื่อใช้ในกระบวนการ Text-to-Speech โดยระบบส่วนนี้จะแสดงผลเป็นเสียงพูดโดยตัวการ์ตูนกราฟิกเคลื่อนไหวที่ชื่อว่า “น้องเทเลคอม” ที่มีลักษณะคล้ายเสาโทรคมนาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.2 ข้อเสนอแนะ

การศึกษาและพัฒนาระบบแนะนำข้อมูลสำหรับภาควิชาชีพวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้สร้างขึ้นมานั้นยังสามารถปรับปรุงแก้ไข และพัฒนาให้มีประสิทธิภาพในอนาคต เพื่อตอบสนองความต้องการของผู้ใช้งานและครอบคลุมการใช้งานให้ครบทุกองค์ประกอบ ซึ่งแบบจำลองยังคงมีข้อบกพร่องอยู่หลายประการ เช่น มีจำนวนชุดข้อมูลที่น้อยเกินไป จึงทำให้แบบจำลองยังคงตอบคำถามได้ไม่ดีมากนัก จึงสามารถแก้ปัญหาได้โดยการเพิ่มชุดข้อมูลให้เหมาะสมกับแบบจำลอง เพื่อให้แบบจำลองสามารถตอบคำถามได้ดียิ่งขึ้น และยังคงมีข้อบกพร่องที่เว็บไซต์ไม่สามารถใช้ได้กับระบบปฏิบัติการบางระบบ เนื่องจากผู้จัดทำยังคงเป็นนักพัฒนาในระดับเบื้องต้นทำให้ยังขาดประสบการณ์ส่วนนี้ จึงสามารถศึกษาการพัฒนาและสร้างเว็บไซต์ให้มากยิ่งขึ้นเพื่อนำไปต่อยอดเพิ่มเติมได้ในอนาคต

บรรณานุกรม

- [1] ผศ.ดร.กอบเกียรติ สระอุบล. เรียนรู้ AI: Deep Learning ด้วย Python. พิมพ์ครั้งที่ 1. กรุงเทพฯ : อินเทอร์เน็ต, 2565
- [2] “Transformer ทำงานยังไง?” <https://huggingface.co/course/th/chapter1/4?fw=pt>.
- [3] ppp mena. “มารู้จัก Transformer กันเถอะ (ตอนที่ 1).” <https://medium.com/mena-ai/%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81-transformer-%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B9%80%E0%B8%96%E0%B8%AD%E0%B8%B0-%E0%B8%95%E0%B8%AD%E0%B8%99%E0%B8%97%E0%B8%B5%E0%B9%88-1-1b99fe798759>.
- [4] ppp mena. “มารู้จัก Transformer กันเถอะ (ตอนที่ 3).” <https://medium.com/mena-ai/%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81-transformer-%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B9%80%E0%B8%96%E0%B8%AD%E0%B8%B0-%E0%B8%95%E0%B8%AD%E0%B8%99%E0%B8%97%E0%B8%B5%E0%B9%88-3-15dcfa97763a>.
- [5] Mehul Gupta. “Attention is all you need: understanding with example.” <https://medium.com/data-science-in-your-pocket/attention-is-all-you-need-understanding-with-example-c8d074c37767>.
- [6] Surapong Kanoktipsatharporn. “Vanishing Gradient Problem คืออะไร แก้ Vanishing Gradient Problem ด้วย Xavier Initialization และ Kaiming Initialization – Neural Network ep.2.” <https://www.bualabs.com/archives/1789/what-is-vanishing-gradient-problem-fix-vanishing-gradient-problem-with-xavier-initialization-kaiming-initialization-neural-network-ep-2/>.

- [7] Pakawat Nakwijit. “ทำความเข้าใจ Transformer [Part I].”
<https://medium.com/@chameleontk/%E0%B8%97%E0%B8%B3%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B9%80%E0%B8%82%E0%B9%89%E0%B8%B2%E0%B9%83%E0%B8%88-transformer-part-i-a50dcf06579b>.
- [8] ppp mena. “มารู้จัก Transformer กันเถอะ (ตอนที่ 2).” <https://medium.com/mena-ai/%E0%B8%A1%E0%B8%B2%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81-transformer-%E0%B8%81%E0%B8%B1%E0%B8%99%E0%B9%80%E0%B8%96%E0%B8%AD%E0%B8%B0-%E0%B8%95%E0%B8%AD%E0%B8%99%E0%B8%97%E0%B8%B5%E0%B9%88-2-7de29224bdc7>.
- [9] Pakawat Nakwijit. “ทำความเข้าใจ BERT.”
<https://medium.com/@chameleontk/%E0%B8%97%E0%B8%B3%E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B9%80%E0%B8%82%E0%B9%89%E0%B8%B2%E0%B9%83%E0%B8%88-bert-98589715545>.
- [10] Jacob Devlin, Ming-Wei Chang, Kenton Lee and Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” <https://arxiv.org/pdf/1810.04805.pdf>.
- [11] Lilian Weng. “How to Build an Open-Domain Question Answering System?” <https://lilianweng.github.io/posts/2020-10-29-odqa/>.
- [12] Rudi Seitz. “UNDERSTANDING TF-IDF AND BM-25.”
<https://kmwllc.com/index.php/2020/03/20/understanding-tf-idf-and-bm-25/>.
- [13] Lan Chu. “Understanding Term-Based Retrieval Methods in Information Retrieval.” <https://towardsdatascience.com/understanding-term-based-retrieval-methods-in-information-retrieval-2be5eb3dde9f>.
- [14] “Reader Models for Open Domain Question-Answering.”
<https://www.pinecone.io/learn/reader-models/>.

- [15] “Preprocess.” <https://huggingface.co/docs/transformers/preprocessing#natural-language-processing>.
- [16] Harshit Tyagi. “The Evolution of Tokenization – Byte Pair Encoding in NLP.” <https://www.freecodecamp.org/news/evolution-of-tokenization/>.
- [17] Isada Sukprapa. “ทำ NER ภาษาไทย โดยใช้ Transformers Model จาก Hugging Face.” <https://medium.com/super-ai-engineer/%E0%B8%97%E0%B8%B3-ner%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2%E0%B9%84%E0%B8%97%E0%B8%A2%E0%B9%82%E0%B8%94%E0%B8%A2%E0%B9%83%E0%B8%8A%E0%B9%89-transformers-model-%E0%B8%88%E0%B8%B2%E0%B8%81-huggingface-130c7a201d5e>.
- [18] VISTEC-depa AI Research Institute of Thailand. “WangchanBERTa โมเดลประมวลผลภาษาไทยที่ใหญ่และก้าวหน้าที่สุดในขณะนี้.” <https://medium.com/airesearch-in-th/wangchanberta-%E0%B9%82%E0%B8%A1%E0%B9%80%E0%B8%94%E0%B8%A5%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B8%A1%E0%B8%A7%E0%B8%A5%E0%B8%9C%E0%B8%A5%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2%E0%B9%84%E0%B8%97%E0%B8%A2%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B9%83%E0%B8%AB%E0%B8%8D%E0%B9%88%E0%B9%81%E0%B8%A5%E0%B8%B0%E0%B8%81%E0%B9%89%E0%B8%B2%E0%B8%A7%E0%B8%AB%E0%B8%99%E0%B9%89%E0%B8%B2%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B8%AA%E0%B8%B8%E0%B8%94%E0%B9%83%E0%B8%99%E0%B8%82%E0%B8%93%E0%B8%B0%E0%B8%99%E0%B8%B5%E0%B9%89-d920c27cd433>.
- [19] “Metric: exact_match.” https://huggingface.co/spaces/evaluate-metric/exact_match.
- [20] Pagon Gatchalee. “Confusion Matrix เครื่องมือสำคัญในการประเมินผลลัพธ์ของการทำนาย ใน Machine learning.” <https://medium.com/@pagongatchalee/confusion-matrix-%E0%B9%80%E0%B8%84%E0%B8%A3%E0%B8%B7%E0%B9%88%E0%B8>

%AD%E0%B8%87%E0%B8%A1%E0%B8%B7%E0%B8%AD%E0%B8%AA%E0%B8%B3%E0%B8%84%E0%B8%B1%E0%B8%8D%E0%B9%83%E0%B8%99%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B9%80%E0%B8%A1%E0%B8%B4%E0%B8%99%E0%B8%9C%E0%B8%A5%E0%B8%A5%E0%B8%B1%E0%B8%9E%E0%B8%98%E0%B9%8C%E0%B8%82%E0%B8%AD%E0%B8%87%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B8%97%E0%B8%B3%E0%B8%99%E0%B8%B2%E0%B8%A2%E0%B9%83%E0%B8%99machine-learning-fba6e3f9508c.

- [21] NIPA Cloud. “Database คืออะไร? แล้ววิวัฒนาการของ Database เป็นอย่างไร?”
<https://nipa.cloud/th/blog/database-clouddatabase>.
- [22] สำนักงานพัฒนาธุรกรรมทางอิเล็กทรอนิกส์. “ศัพท์ชว่นรู้.”
<https://www.etcha.or.th/th/Useful-Resource/terminology/%E0%B8%AB%E0%B8%A1%E0%B8%A7%E0%B8%94%E0%B8%AB%E0%B8%A1-D/221.aspx>
- [23] “องค์ประกอบของระบบฐานข้อมูล.”
<http://academic.udru.ac.th/~samawan/content/6sa-DBDesign.pdf>.
- [24] “Database คืออะไร ระบบฐานข้อมูล คือ การเก็บรวบรวมข้อมูลไว้ด้วยกัน อย่างมีระบบ.”
<https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/2055-database-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A323.html>.
- [25] ศูนย์เทคโนโลยีสารสนเทศและการสื่อสาร สป. “บทที่ 7 ความรู้พื้นฐานเกี่ยวกับ MySQL.”
http://elearning.psu.ac.th/courses/66/SQL/CH07_BasicMySQL.pdf.
- [26] “MYSQL คืออะไร?” <https://www.websiterating.com/th/web-hosting/glossary/what-is-mysql/>.
- [27] “ภาษา SQL คืออะไร.”

<https://www.9experttraining.com/articles/%E0%B8%A0%E0%B8%B2%E0%B8%A9%E0%B8%B2-sql-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3>

- [28] Developer Team BorntoDev Co., Ltd. “Microservices คืออะไร ใช้ยังไง ?.”
<https://www.borntodev.com/2020/05/22/microservices-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/>.
- [29] SEOThailand. “HTTP และ HTTPS คืออะไร และ แตกต่างกันอย่างไรร.”
<https://www.seothailand.in.th/http-and-https/>.
- [30] “เว็บไซต์ (Website) ความหมายของเว็บไซต์ ประโยชน์ที่สำคัญ และองค์ประกอบต่างๆ.”
<https://www.1belief.com/article/website/>.
- [31] นางสาวจิรวรรณ ถึงไชย, นางสาวสุธิตา สีตาแสน. “หน่วยที่ 1 พื้นฐานและการทำงานของบริการเว็บบนระบบอินเทอร์เน็ต.”
<https://sites.google.com/site/www33120sutitaseetasan/hnwy-thi-1>.
- [32] Jenna Inouye. “What is a Web Server and How Do They Work? Ultimate Guide [2023].” <https://hackr.io/blog/what-is-a-web-server>.
- [33] Kriangkrai Chaonithi. “ทำเว็บต้องรู้อะไรบ้าง? เอากันตั้งแต่คอมพิวเตอร์ทำงานยังไง ไปจนถึงการสร้างเว็บเอาขึ้นออนไลน์.” <https://www.spicydog.org/blog/what-is-web-application/>.
- [34] “เว็บเบราว์เซอร์.”
<https://th.wikipedia.org/wiki/%E0%B9%80%E0%B8%A7%E0%B9%87%E0%B8%9A%E0%B9%80%E0%B8%9A%E0%B8%A3%E0%B8%B2%E0%B8%A7%E0%B9%8C%E0%B9%80%E0%B8%8B%E0%B8%AD%E0%B8%A3%E0%B9%8C>.
- [35] “HTML คืออะไร เอชทีเอ็มแอล ภาษาคอมพิวเตอร์ที่ใช้ในการสร้างเว็บเพจ ใช้เขียนโปรแกรมย่อมาจากอะไร.”
<https://www.mindphp.com/%E0%B8%84%E0%B8%B9%E0%B9%88%E0%B8%A1%E0%B8%B7%E0%B8%AD/73->

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B
9%84%E0%B8%A3/2026-html-

%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B
9%84%E0%B8%A3.html.

[36] พิชิต วิจิตรบุญรักษ์. “HTML: ภาษาเขียนเว็บ HTML: A Web Language.”

https://www.bu.ac.th/knowledgecenter/executive_journal/july_sep_11/pdf/aw32.pdf.

[37] “บทที่ 1 HTML5 คืออะไร เอชทีเอ็มแอลห้า คือ ภาษามาร์กอัป ที่ใช้สำหรับเขียน website ที่พัฒนามาจาก HTML.”

<https://www.mindphp.com/%E0%B8%9A%E0%B8%97%E0%B9%80%E0%B8%A3%E0%B8%B5%E0%B8%A2%E0%B8%99%E0%B8%AD%E0%B8%AD%E0%B8%99%E0%B9%84%E0%B8%A5%E0%B8%99%E0%B9%8C/%E0%B8%9A%E0%B8%97%E0%B9%80%E0%B8%A3%E0%B8%B5%E0%B8%A2%E0%B8%99-html5/2479->

<https://www.mindphp.com/%E0%B8%9A%E0%B8%97%E0%B8%97%E0%B8%B5%E0%B9%88-1-html5-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A331.html>.

[38] “CSS คืออะไร ? มีประโยชน์อย่างไรบ้าง.” <https://www.wynnsolution.net/th/article/view/80/>.

[39] Kipakapron. “CSS คืออะไร มีประโยชน์ อย่างไร.”

<https://blog.sogoodweb.com/Article/Detail/79237/CSS->

<https://blog.sogoodweb.com/Article/Detail/79237/CSS-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3->

<https://blog.sogoodweb.com/Article/Detail/79237/CSS-%E0%B8%A1%E0%B8%B5%E0%B8%9B%E0%B8%A3%E0%B8%B0%E0%B9%82%E0%B8%A2%E0%B8%8A%E0%B8%99%E0%B9%8C->

<https://blog.sogoodweb.com/Article/Detail/79237/CSS-%E0%B8%AD%E0%B8%A2%E0%B9%88%E0%B8%B2%E0%B8%87%E0%B9%84%E0%B8%A3>.

[40] “JAVASCRIPT คืออะไร ความรู้เบื้องต้นที่ควรรู้.” <https://allweb.co.th/159->

<https://allweb.co.th/159-%E0%B8%9A%E0%B8%A5%E0%B8%AD%E0%B8%84/1297-javascript->

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-
 %E0%B8%84%E0%B8%A7%E0%B8%B2%E0%B8%A1%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B9%80%E0%B8%9A%E0%B8%B7%E0%B9%89%E0%B8%AD%E0%B8%87%E0%B8%95%E0%B9%89%E0%B8%99%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B8%84%E0%B8%A7%E0%B8%A3%E0%B8%A3%E0%B8%B9%E0%B9%89.html.

[41] Frame. “Django Cloudinary เก็บไฟล์ media ขึ้น Cloud ด้วย Cloudinary.”

[https://stackpython.co/tutorial/django-cloudinary-media-cloud-cloudinary.](https://stackpython.co/tutorial/django-cloudinary-media-cloud-cloudinary)

[42] “web server พื้นฐานเบื้องต้นของการใช้งานเว็บไซต์ที่หลายคนยังไม่รู้.”

[https://www.quickserve.co.th/knowledge-base/solutions/web-server-%E0%B8%9E%E0%B8%B7%E0%B9%89%E0%B8%99%E0%B8%90%E0%B8%B2%E0%B8%99%E0%B9%80%E0%B8%9A%E0%B8%B7%E0%B9%89%E0%B8%AD%E0%B8%87%E0%B8%95%E0%B9%89%E0%B8%99%E0%B8%82%E0%B8%AD%E0%B8%87%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B9%80%E0%B8%A7%E0%B9%87%E0%B8%9A%E0%B9%84%E0%B8%8B%E0%B8%95%E0%B9%8C%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B8%AB%E0%B8%A5%E0%B8%B2%E0%B8%A2%E0%B8%84%E0%B8%99%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B9%84%E0%B8%A1%E0%B9%88%E0%B8%A3%E0%B8%B9%E0%B9%89/.](https://www.quickserve.co.th/knowledge-base/solutions/web-server-%E0%B8%9E%E0%B8%B7%E0%B9%89%E0%B8%99%E0%B8%90%E0%B8%B2%E0%B8%99%E0%B9%80%E0%B8%9A%E0%B8%B7%E0%B9%89%E0%B8%AD%E0%B8%87%E0%B8%95%E0%B9%89%E0%B8%99%E0%B8%82%E0%B8%AD%E0%B8%87%E0%B8%81%E0%B8%B2%E0%B8%A3%E0%B9%83%E0%B8%8A%E0%B9%89%E0%B8%87%E0%B8%B2%E0%B8%99%E0%B9%80%E0%B8%A7%E0%B9%87%E0%B8%9A%E0%B9%84%E0%B8%8B%E0%B8%95%E0%B9%8C%E0%B8%97%E0%B8%B5%E0%B9%88%E0%B8%AB%E0%B8%A5%E0%B8%B2%E0%B8%A2%E0%B8%84%E0%B8%99%E0%B8%A2%E0%B8%B1%E0%B8%87%E0%B9%84%E0%B8%A1%E0%B9%88%E0%B8%A3%E0%B8%B9%E0%B9%89/)

[43] ปวรุตม์ พงศ์พัฒนานนท์. “Framework คืออะไร.”

[https://km.cc.swu.ac.th/archives/4235.](https://km.cc.swu.ac.th/archives/4235)

[44] Amalia Taylor. “WebSockets คืออะไรและจะสร้างได้อย่างไร.”

[https://appmaster.io/th/blog/websockets-khuue-aairaelacchasraangaid-yaangair.](https://appmaster.io/th/blog/websockets-khuue-aairaelacchasraangaid-yaangair)

[45] จิตกร พิทักษ์เมธากุล. “WebSocket คืออะไร ทำงานยังไง (อธิบายแบบละเอียด).”

[https://www.jittagornp.me/blog/what-is-websocket/.](https://www.jittagornp.me/blog/what-is-websocket/)

[46] สรรพโชค สิงหสุวรรณ. “การใช้งาน Docker เบื้องต้น.”

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

https://race.nSTRU.ac.th/home_ex/blog/topic/show/4200.

[47] “Docker คืออะไร?”

https://computer.ru.ac.th/images/Km2Article/1659670212_Article.pdf.

[48] Patcharaporn Auerach. “เปลืองเวลาเป็นวัน ๆ ไปทำไม ? ในเมื่อใช้ Docker ช่วยได้ !.”

<https://blog.openlandscape.cloud/docker>.

[49] Piotr Golofit. “What is Docker and why to use it? Explained for executives.”

<https://accesto.com/blog/what-is-docker-and-why-to-use-it/>.

[50] “รู้จักกับ Docker แพลตฟอร์มซอฟต์แวร์สำหรับนักพัฒนาและผู้ดูแลระบบ.”

<https://www.quickserve.co.th/knowledge-base/solutions/%E0%B8%A3%E0%B8%B9%E0%B9%89%E0%B8%88%E0%B8%B1%E0%B8%81%E0%B8%81%E0%B8%B1%E0%B8%9A-Docker-%E0%B9%81%E0%B8%9E%E0%B8%A5%E0%B8%95%E0%B8%9F%E0%B8%AD%E0%B8%A3%E0%B9%8C%E0%B8%A1%E0%B8%8B%E0%B8%AD%E0%B8%9F%E0%B8%95%E0%B9%8C%E0%B9%81%E0%B8%A7%E0%B8%A3%E0%B9%8C%E0%B8%AA%E0%B8%B3%E0%B8%AB%E0%B8%A3%E0%B8%B1%E0%B8%9A%E0%B8%99%E0%B8%B1%E0%B8%81%E0%B8%9E%E0%B8%B1%E0%B8%92%E0%B8%99%E0%B8%B2%E0%B9%81%E0%B8%A5%E0%B8%B0%E0%B8%9C%E0%B8%B9%E0%B9%89%E0%B8%94%E0%B8%B9%E0%B9%81%E0%B8%A5%E0%B8%A3%E0%B8%B0%E0%B8%9A%E0%B8%9A/>.

[51] Thanatip Suwanjandee. “Google Cloud คืออะไร (ฉบับมือใหม่).” <https://cloud-ace.co.th/blogs/o6t7k4-google-cloud>.

[52] “Google Cloud products.” <https://cloud.google.com/products>.

[53] Surapong Kanoktipsatharporn. “Heroku คืออะไร สอน Heroku ภาษาไทย เริ่มต้นสมัครใช้งาน Heroku PaaS – heroku ep.1.”

<https://www.bualabs.com/archives/3916/what-is-heroku-tutorial-heroku-paas-in-thai-language-how-to-build-app-dyno-in-python-heroku-ep-1/>.

[54] Onamon Ja. “Heroku คืออะไร?” <https://medium.com/cscmu-undergrad-seminar/heroku->

%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-fdc183e9af47.

[55] Pakin Phuhinkong. “Git คืออะไร ... Git is your friend.”

<https://medium.com/@pakin/git->

%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-git-is-your-friend-c609c5f8efea.

[56] “Fast tokenizers in the QA pipeline.”

[https://huggingface.co/course/chapter6/3b?fw=pt.](https://huggingface.co/course/chapter6/3b?fw=pt)

[57] “Optuna: A hyperparameter optimization framework.”

[https://github.com/optuna/optuna.](https://github.com/optuna/optuna)

[58] “WangchanBERTa: Pre-trained Thai Language Model.”

[https://airesearch.in.th/releases/wangchanberta-pre-trained-thai-language-model/.](https://airesearch.in.th/releases/wangchanberta-pre-trained-thai-language-model/)

[59] “Fine-tune a pretrained model.”

[https://huggingface.co/docs/transformers/v4.24.0/en/training.](https://huggingface.co/docs/transformers/v4.24.0/en/training)

[60] “Trainer.”

[https://huggingface.co/docs/transformers/main/en/main_classes/trainer#transformers.](https://huggingface.co/docs/transformers/main/en/main_classes/trainer#transformers)

[60] “vistec-AI/thai2transformers.” [https://github.com/vistec-AI/thai2transformers.](https://github.com/vistec-AI/thai2transformers)

[61] “Google Collab หรือ Google Colaboratory: มันคืออะไร.”

<https://www.hwlibre.com/th/google-colaboratory/>

[62] ดร.ณัฐพล แสนคำ. “วิธีการใช้งาน Visual Studio Code.”

<https://cs.bru.ac.th/%E0%B8%AA%E0%B8%AD%E0%B8%99%E0%B8%A7>

%E0%B8%B4%E0%B8%98%E0%B8%B5%E0%B8%81%E0%B8%B2%E0%B8

%A3%E0%B9%83%E0%B8%8A%E0%B9%89-visual-studio-code-2/.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def _get_context_span(input_ids,
                      sequence_ids,
                      answers,
                      start_col,
                      text_col=TEXT_COL,
                      pad_on_right=True):
    # Start/end character index of the answer in the text.
    start_char = answers[start_col][0]
    end_char = start_char + len(answers[text_col][0]) + 1
    # Start token index of the current span in the text.
    token_start_index = 0
    while sequence_ids[token_start_index] != (1 if pad_on_right else 0):
        token_start_index += 1
    # End token index of the current span in the text.
    token_end_index = len(input_ids) - 1
    while sequence_ids[token_end_index] != (1 if pad_on_right else 0):
        token_end_index -= 1
    return token_start_index, token_end_index, start_char, end_char

def _get_answer_span(tokenized_examples,
                     offsets,
                     start_char,
                     end_char,
                     token_start_index,
                     token_end_index,):
    while token_start_index < len(offsets) and offsets[token_start_index][0] <=
start_char:
        token_start_index += 1
    start_position = token_start_index - 1
    while offsets[token_end_index][1] >= end_char:
        token_end_index -= 1
    end_position = token_end_index + 1

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

return start_position, end_position
def prepare_qa_train_features(examples,
                             tokenizer,
                             question_col='question',
                             context_col='context',
                             answers_col='answers',
                             start_col=START_COL,
                             text_col=TEXT_COL,
                             pad_on_right=True,
                             max_length=416,
                             doc_stride=128):
    tokenized_examples = tokenizer(
        examples[question_col],
        examples[context_col],
        truncation="only_second",
        max_length=max_length,
        stride=doc_stride, #overlapping of overflowing tokens
        return_overflowing_tokens=True, #return multiple input ids if exceeding
max_length
        return_offsets_mapping=True,
        padding="max_length",
    )
    sample_mapping = tokenized_examples.pop("overflow_to_sample_mapping")
#map overflowing examples to original examples
    print(sample_mapping)
    offset_mapping = tokenized_examples.pop("offset_mapping") #offset map;
character index not resetted across overflowing examples
    tokenized_examples["start_positions"] = []
    tokenized_examples["end_positions"] = []
    #loop through all examples' offset_mapping
    for i, offsets in enumerate(offset_mapping):

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

input_ids = tokenized_examples["input_ids"][i] #a list of tokens
cls_index = input_ids.index(tokenizer.cls_token_id) #use cls as answer when
there is no answer
sequence_ids = tokenized_examples.sequence_ids(i) #sequence_ids 0 for
question and 1 for context
sample_index = sample_mapping[i] #since many examples can point to the
same question due to overflowing
answers = examples[answers_col][sample_index] #answers of each example
# If no answers are given, set the cls_index as answer
if len(answers[text_col]) == 0:
    tokenized_examples["start_positions"].append(cls_index)
    tokenized_examples["end_positions"].append(cls_index)
else:
    token_start_index, token_end_index, start_char, end_char =
_get_context_span(input_ids=input_ids,
                    sequence_ids=sequence_ids,
                    answers=answers,
                    start_col=start_col,
                    text_col=text_col,
                    pad_on_right=pad_on_right)
    # If answer is not in span, return cls_index as answer
    if not (offsets[token_start_index][0] <= start_char and
offsets[token_end_index][1] >= end_char-1):
        tokenized_examples["start_positions"].append(cls_index)
        tokenized_examples["end_positions"].append(cls_index)
    else:
        start_position, end_position = _get_answer_span(tokenized_examples,
            offsets,
            start_char,
            end_char,
            token_start_index,

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
token_end_index,)
tokenized_examples["start_positions"].append(start_position)
tokenized_examples["end_positions"].append(end_position)
return tokenized_examples
tokenized_datasets = datasets.map(lambda x: prepare_qa_train_features(x, tokenizer),
                                  batched=True,
                                  remove_columns=datasets["train"].column_names)
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

model =
AutoModelForQuestionAnswering.from_pretrained("airesearch/wangchanberta-base-
att-spm-uncased")
batch_size = 16
learning_rate = 8e-5
args = TrainingArguments(
    f"finetune_telecom_thaiqa",
    evaluation_strategy = "epoch",
    learning_rate=learning_rate,
    per_device_train_batch_size=batch_size,
    per_device_eval_batch_size=batch_size,
    num_train_epochs=3,
    weight_decay=0.00013,
    save_steps=2000,
    fp16=True,
    #adam_epsilon = 1e-8,
    #adam_beta1 = 0.9,
    #adam_beta2 = 0.999,
    logging_strategy = "epoch",
)
data_collator = default_data_collator
trainer = Trainer(
    model=model,
    args=args,
    train_dataset=tokenized_datasets["train"],
    eval_dataset=tokenized_datasets["validation"],
    data_collator=data_collator,
    tokenizer=tokenizer,
)
trainer.train()
trainer.save_model("/content/finetune_telecom_thaiqa/finetune_newdataset5")

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

โปรแกรมของระบบแนะนำข้อมูลสำหรับภาควิชาวิศวกรรมโทรคมนาคม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

from transformers import AutoModelForQuestionAnswering, AutoTokenizer, pipeline
from pythainlp import word_tokenize, Tokenizer
from pythainlp.corpus import thai_stopwords
from rank_bm25 import BM25Okapi
import numpy as np
import mysql.connector
import requests # test
from mysql.connector import errorcode
import os

class Database:
    def __init__(self):
        self.host = "35.186.147.213"
        self.user="root"
        self.password="29062544"
        self.database="telecom_qa_db"
        self.mycursor = None
        self.mydb = None
    def set_host(self, host):
        self.host = host
    def set_user(self, user):
        self.user = user
    def set_password(self, password):
        self.password = password
    def set_database(self, database):
        self.database = database
    def get_context_from_db(self):
        mydb = mysql.connector.connect(host = self.host, user = self.user, passwd =
self.password, database = self.database)
        self.mydb = mydb
        mycursor = mydb.cursor()
        self.mycursor = mycursor

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

mycursor.execute("SELECT * FROM telecom_qa_db.context_info")
myresult = mycursor.fetchall()
# print(myresult)
# Grab only context
cntx = [dt[1] for dt in myresult]
# mycontext = []
# cntx = []
# # Grab context and created_at
# for data in myresult:
#     mycontext.append((data[1],data[2]))
# # Grab only context
# for dt in mycontext:
#     cntx.append(dt[0].lower()) #lastest
return cntx
def insert_unanswerable_question_to_db(self, question):
print("Logging : insert_unanswerable_question_to_db")
try:
    sql = """INSERT INTO unanswerable_question (question) VALUES (%s)"""
    val = [(question)]
    self.mycursor.execute(sql, val)
    self.mydb.commit()
    print(self.mycursor.rowcount, "Record inserted successfully into table")
except mysql.connector.Error as error:
    print("Failed to insert into MySQL table {}".format(error))
# finally:
#     if self.mydb.is_connected():
#         self.mycursor.close()
#         self.mydb.close()
#         print("MySQL connection is closed")
class QAReaderModel:

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

def __init__(self, model_path=
os.getcwd()+"/app/model/finetune_11_03_2023_multilinguamodel-round3-
hugedataset"):
    """_summary_
    Args:
        model_path (str): Directory containing the necessary tokenizer and model
files.
    """
    self.model = None
    self.tokenizer = None
    self.model_path = model_path
    # self.qa = None
    self.set_model(model_path)
def qa(self):
    qa = pipeline("question-answering", model=self.model, tokenizer=self.tokenizer)
    return qa
# def get_qa(self, model, tokenizer):
#     qa = pipeline("question-answering", model=model, tokenizer=tokenizer)
#     return qa
# def set_qa(self):
#     self.qa = self.get_qa(self.model, self.tokenizer)
def get_model(self, model_path):
    """
    Load a tokenizer and model using 'AutoTokenizer' and 'Automodel'
    Args:
        model_path (str): Directory containing the necessary tokenizer and model
files.
    """
    model = AutoModelForQuestionAnswering.from_pretrained(model_path)
    print("modellllllll : ", model)
    # model = AutoModelForQuestionAnswering.from_pretrained(model_path)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

tokenizer = AutoTokenizer.from_pretrained(model_path)
print("tokenizerrrrrrrr : ", tokenizer)
return model, tokenizer
def set_model(self, model_path):
    """
    Set a tokenizer and model using the 'self.get_model' method.
    Args:
        model_path (str): Directory containing the necessary tokenizer and model
files.
    """
    self.model, self.tokenizer = self.get_model(model_path)
def showData(self):
    # print(self.model, "\n", self.tokenizer, "\n", self.qa)
    # print(self.qa)
    qa = self.qa()
    print("qa info : {}".format(qa))
class QARetriever:
    def __init__(self,
model_path=os.getcwd()+"/app/model/finetune_11_03_2023_multilinguamodel-
round3-hugedataset"):
        """
        Define a QA Retriever model.
        Search for the associated context and return the best answer.
        Args:
            model_path (str): Directory containing the necessary tokenizer and model
files.
        """
        self.answer = None
        self.question = None # query question
        self.greeting = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

self.preprocessed_question = None # preprocessed query question
self.contexts = None
self.preprocessed_contexts = None
self.qa_reader = QAReaderModel(model_path=model_path)
self.tokenizer = self.qa_reader.tokenizer
self.model = None
self.qa = self.qa_reader.qa()
self.database = Database() ##
# self.best_answer = None # best score w/ best answer
self.condition = False
def set_question(self, question): # with api part
    self.question = question
def set_preprocessed_question(self):
    self.preprocessed_question = self.preprocess_query(self.question)
def set_greeting(self):
    greeting = ['สวัสดีครับ', 'สวัสดีค่ะ', 'สวัสดี', 'หวัดดีครับ', 'หวัดดีค่ะ', 'หวัดดี', 'Hi', 'hi', 'hello',
'Hello', 'ฮัลโหล', 'สวัสดีจ้า', 'ดีฮะ', 'ดีจ้า']
    just_greeting = False
    if str([self.question]) == str([val for val in greeting if self.question == val]):
        just_greeting = True
    else:
        pass
    self.greeting = just_greeting
def set_preprocessed_contexts(self):
    self.preprocessed_contexts = self.preprocess_contexts()
    ##logging
    # print("preprocessed_contexts (in set_contexts) :
    {}".format(self.preprocessed_contexts))
def preprocess_query(self, question):
    # start changing
    token_question = self.tokenizer(question).input_ids

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

token_question = token_question[1:-1:] #remove special token
decode_question = []
for x in token_question:
    if x == 10: #Remove white space (token 10)
        pass
    else:
        decode = self.tokenizer.decode(x)
        decode_question.append(decode)
already_question = decode_question
# print("Token question: ", already_question)
# Remove Stopwords
stopwords = list(thai_stopwords())
already_question = [i for i in decode_question if i not in stopwords]
# print("Already Question: ", already_question)
# end changing
## Tokenization
# token_question = word_tokenize(question, engine="newmm",
keep_whitespace=False)
# Remove stopwords
# stopwords = list(thai_stopwords())
return already_question

def preprocess_contexts(self):
    all_context = self.database.get_context_from_db()
    # Set contexts obtained database
    self.contexts = all_context
    # start changing
    already_context = []
    for each_cntx in all_context:
        decode_context = []
        token_contexts = self.tokenizer(each_cntx.lower()).input_ids
        token_contexts = token_contexts[1:-1:] # remove special token

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for x in token_contexts:
    if x == 10: # white space removing (token 10)
        pass
    else:
        decode = self.tokenizer.decode(x)
        decode_context.append(decode)
    already_context.append(decode_context)
    del decode_context
# end changing
# token_contexts = []
# for context in all_context:
#     token_contexts.append(word_tokenize(context, engine="newmm",
keep_whitespace=False))
# ## Logging
# # print("token_contexts : {}".format(token_contexts))
# return token_contexts
return already_context

def retriever(self):
    # Initializing
    bm25 = BM25Okapi(self.preprocessed_contexts)
    # Ranking of documents
    doc_scores = bm25.get_scores(self.preprocessed_question)
    # Getting top 3 relevant context
    ranked_contexts = bm25.get_top_n(self.preprocessed_question, self.contexts,
n=1)

    # Max to min bm25 score sorting
    top_n_score = np.flip(np.sort(doc_scores))
    # # Display ranking score
    # print("\nTop 3 most similar sentences in corpus:\n")
    # for score, ct in zip(top_n_score, ranked_contexts):
    #     print("(Score: {:.4f})".format(score),ct)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# print("\n\n")
return ranked_contexts
def get_best_answers(self):
    # start adding
    self.condition_checking()
    if self.condition == True: # found eng
        self.change_model()
    else:
        self.default_model()
    # stop adding
    self.set_preprocessed_contexts()
    self.set_preprocessed_question()

    ranked_contexts = self.retriever()
    answer = []
    # print("context in ranked_contexts")
    for context in ranked_contexts:
        ans = self.qa(question=self.question, context=context)
        answer.append((ans["score"], ans["answer"]))
    # Max to min answer score
    answer.sort(reverse=True)
    best_answer = max(answer) # score with answer
    # # testing start
    # if self.condition == True:
    #     API_URL = "https://api-inference.huggingface.co/models/wicharnkeisei/thai-
xml-roberta-base-squad2"
    #     headers = {"Authorization": f"Bearer
{'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'}"}
    #     def query(payload):
    #         response = requests.post(API_URL, headers=headers, json=payload)
    #         return response.json()

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

# for context in ranked_contexts:
#     output = query({
#         "inputs": {
#             "question": "{}".format(self.question),
#             "context": "{}".format(context)
#         },
#     })
#     answer.append((output['score'], output['answer']))
# answer.sort(reverse=True)
# best_answer = max(answer) # score with answer
# else:
#     for context in ranked_contexts:
#         ans = self.qa(question=self.question, context=context)
#         answer.append((ans["score"], ans["answer"]))
# # Max to min answer score
# answer.sort(reverse=True)
# best_answer = max(answer) # score with answer
# # testing end
return best_answer

def get_answer(self, question):
    self.set_question(question=question)
    # start changing
    self.set_greeting()
    self.condition = False
    if self.greeting == True:
        answer = "สวัสดี ยินดีให้บริการ"
    else:
        best_answer = self.get_best_answers()
        softmax_score = best_answer[0]
        print("softmaxxxxxxxx: ",softmax_score)
        threshold = 0.3

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if softmax_score < threshold:
    # print("Loggingggggggg_")
    # insert unanswerable question to db
    self.database.insert_unanswerable_question_to_db(question)
    answer = "ขออภัย ไม่สามารถตอบคำถามนี้ได้"
else:
    answer = best_answer[1] # best text answer
# end changing
return answer
def default_model(self):
    self.tokenizer = self.qa_reader.tokenizer
    self.model = self.qa_reader.model
    self.qa = self.qa_reader.qa()
def change_model(self, model_path="wicharnkeisei/thai-xlm-roberta-base-squad2"):
    self.set_changed_model(model_path)
    self.set_changed_qa()
def get_changed_model(self, model_path):
    model = AutoModelForQuestionAnswering.from_pretrained(model_path)
    tokenizer = AutoTokenizer.from_pretrained(model_path)
    return model, tokenizer
def set_changed_model(self, model_path):
    self.model, self.tokenizer = self.get_changed_model(model_path)
def get_changed_qa(self):
    qa = pipeline("question-answering", model=self.model, tokenizer=self.tokenizer)
    return qa
def set_changed_qa(self):
    self.qa = self.get_changed_qa()
def condition_checking(self):
    engs = ['ภาษาอังกฤษ', 'อิง', 'อังกฤษ']
    tokenize_input = word_tokenize(self.question)
    condition = False

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
for i in engs:
    for j in tokenize_input:
        if j == i:
            condition = True
            # print(i,j)
            pass
        self.set_condition(condition)
def set_condition(self, condition):
    self.condition = condition
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ง

โปรแกรมสำหรับการสร้าง Web Service API

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

import uvicorn
from fastapi import FastAPI, Request
from pydantic import BaseModel
from app.utils import QARetriever
app = FastAPI()
qa_search = QARetriever()
class TextIn(BaseModel):
    text: str
class PredictionOut(BaseModel):
    answer: str
@app.get("/")
def home():
    return {"status_health_check": "OK"}
@app.post("/predict", response_model=PredictionOut)
def predict(payload: TextIn):
    answer = qa_search.get_answer(payload.text)
    return {"answer" : answer}
# # initializes the QA model and start the uvicorn app
# if __name__ == "__main__":
#     qa_search = QARetriever()
#     uvicorn.run(app, host="0.0.0.0", port=8000)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก จ

โปรแกรมที่ใช้ในการสร้าง Container Image (Dockerfile)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
FROM python:3.9
LABEL author="Patcharakit Yangdiaw"
#
WORKDIR /code
#
COPY ./requirements.txt /code/requirements.txt
#
RUN pip install --no-cache-dir --upgrade -r /code/requirements.txt
#
COPY ./app /code/app
EXPOSE 8000
#
CMD ["uvicorn", "app.main:app", "--host", "0.0.0.0", "--reload"]
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Telecomm Chan</title>
  <!-- <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-
alpha1/dist/css/bootstrap.min.css" rel="stylesheet" -->
  <!-- integrity="sha384-
GLhLTQ8iRABdZLL6O3oVMWSktQOp6b7In1ZL3/Jr59b6EGGol1aFkw7cmDA6j6gD"
crossorigin="anonymous"> -->
  <link rel="icon" type="image/x-icon"
href="https://res.cloudinary.com/dgumpoqcu/image/upload/v1676825597/png_fy5gzk.
png">
  <link rel="style">
<style>
  html {
    font-family: 'Mitr', sans-serif;
    box-sizing: border-box;
    -webkit-text-size-adjust: 100%;
  }
  p {
    margin: 0.25rem;
  }
  h1 {
    font-family: 'Mitr', sans-serif;
    font-size: 18px;
    text-align: center;
    color: #000000;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
h3 {
  font-size: 16px;
  font-weight: 550;
  text-align: center;
  color: #ffffff;
}
body {
  font-family: 'Mitr', sans-serif;
  -ms-word-wrap: break-word;
  word-wrap: break-word;
  background-color: #666666;
  background-image:
url('https://res.cloudinary.com/dgumpoqcu/image/upload/v1676825597/telecomweb_
mkwocq.jpg');
  background-position: center top;
  background-size: cover;
  background-origin: adding-box;
  background-repeat: no-repeat;
  position: fixed;
}
.open-button {
  cursor: pointer;
  opacity: 1;
  position: fixed;
  bottom: 3%;
  right: 1%;
  width: 110px;
  filter: contrast(110%);
}
.open-button .btn {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

background-color: transparent;
border: none;
cursor: pointer;
}
.open-button .btn:hover {
  /*background-color: #FA4616;*/
  background-color: #E35205;
  border-radius: 32px;
  animation-name: telecomchan;
  animation-duration: 0.6s;
}
@keyframes telecomchan {
  0% {
    transform: rotate(0);
  }
  20% {
    transform: rotate(15deg);
  }
  40% {
    transform: rotate(-15deg);
  }
  60% {
    transform: rotate(0);
  }
  80% {
    transform: rotate(15deg);
  }
  100% {
    transform: rotate(0);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

@media only screen and (max-width: 768px) and (min-width: 320px) {
  .chat-popup {
    height: 50%;
    width: 100%;
    bottom: 0;
    right: 0;
  }
  .header {
    width: 50%;
  }
  .telecomchan {
    width: 60%;
    height: 30%;
    bottom: 25%;
  }
  .show-input {
    bottom: 75px;
  }
}
@media only screen and (max-width: 1365px) and (min-width: 769px) {
  .chat-popup {
    height: 49%;
    width: 45%;
    bottom: 3%;
    right: 1%;
  }
  .telecomchan {
    width: 90%;
    height: 40%;
    bottom: 25%;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

.show-input {
    bottom: 100px;
}
}
@media only screen and (min-width: 1366px) {
.chat-popup {
    /* height: 42%; */
    height: 50%;
    /* width: 27%; */
    width: 27%;
    bottom: 3.5%;
    right: 1%;
}
.telecomchan {
    width: 80%;
    height: 40%;
    bottom: 25%;
}
.show-input {
    /*bottom: 80px;*/
    bottom: 120px;
}
}
.chat-popup {
    display: -webkit-Flex;
    background-color: #FF9270;
    display: none;
    grid-template-rows: 10% 40% 20% 30%;
    position: fixed;
    border: none;
    backdrop-filter: blur(10px);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    box-sizing: border-box;
    border-radius: 32px;
}
.header {
  top: 0;
  right: 0;
  background: #FA4616;
  padding: 0.15rem;
  position: fixed;
  display: flex;
  height: 2.6rem;
  box-sizing: border-box;
  backdrop-filter: blur(10px);
  border-radius: 32px;
}
.how2use-btn {
  background-color: #1d3557;
  padding-left: 15px;
  padding-right: 15px;
  font-size: medium;
  text-align: center;
  flex-grow: 1;
  border-radius: 32px;
  margin: 0.2rem;
  cursor: pointer;
  /*opacity: 0.85;*/
  border: none;
  color: white;
}
.cancel-btn {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

text-align: center;
background-color: #ffc1c5;
/*padding: 0.3rem;*/
border-radius: 100%;
margin: 0.1rem;
cursor: pointer;
border: none;
filter: contrast(150%);
}
.cancel-btn:hover {
background-color: white;
}
.how2use-btn:hover {
background-color: white;
color: #003A6A;
}
.display-show-output {
position: fixed;
display: flex;
box-sizing: border-box;
margin: 0% 10% 0 0;
}
/* width */
.show-output::-webkit-scrollbar {
width: 13px;
}
/* Track */
.show-output::-webkit-scrollbar-track {
box-shadow: inset 0 0 5px grey;
border-radius: 10px 80px 80px 10px;
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

/* Handle */
.show-output::-webkit-scrollbar-thumb {
    background: #1d3557;
    border-radius: 10px 80px 80px 10px;
}

/* Handle on hover */
.show-output::-webkit-scrollbar-thumb:hover {
    background: #1d3557c7;
}

.show-output {
    flex-grow: 1;
    position: fixed;
    background-color: rgb(255, 255, 255);
    color: black;
    padding: 15px 15px;
    border-radius: 30px 25px 25px 0px;
    box-shadow: 5px 5px 15px 2px rgba(0, 0, 0, 0.2);
    width: fit-content;
    text-overflow: clip;
    overflow-y: auto;
    word-break: break-word;
    text-align: start;
    font-family: 'Mitr', sans-serif;
    font-size: 16px;
    margin-top: 15%;
    margin-left: 39%;
    margin-right: 10%;
    line-height: 1.55;
    height: fit-content;
}

.show-input {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

flex-grow: 1;
display: none;
position: fixed;
background-color: white;
color: black;
padding: 15px 15px;
border-radius: 25px 30px 0px 25px;
box-shadow: 5px 5px 15px 2px rgba(0, 0, 0, 0.2);
width: fit-content;
text-overflow: clip;
overflow: auto;
word-break: break-word;
text-align: end;
font-family: 'Mitr', sans-serif;
font-size: 16px;
margin-top: 13%;
margin-left: 33%;
line-height: 1.55;
right: 9%;
}
.showTelecomchan {
width: 100%;
height: 100%;
position: fixed;
margin-top: 20%;
display: flex;
}
.telecomchan {
background-image:
url('https://res.cloudinary.com/dgumpoqcu/image/upload/v1678897044/hi1_0000_1_lukgiv.gif');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

background-position: left;
background-size: contain;
background-repeat: no-repeat;
filter: contrast(110%);
position: fixed;
flex-grow: 1;
box-sizing: border-box;
}
.footer {
bottom: 0;
left: 0;
right: 0;
background: #FA4616;
padding: 0.25rem;
position: fixed;
display: flex;
height: 3.5rem;
box-sizing: border-box;
backdrop-filter: blur(100px);
border-radius: 28px;
width: 100%;
}

#form {
flex-grow: 1;
box-sizing: border-box;
display: flex;
width: 100%;
}

#input {
flex-grow: 1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

border: none;
border-radius: 2rem;
margin: 0.25rem;
}
#input:focus {
outline: none;
padding: 0 1rem;
}
#form>button {
padding-left: 20px;
padding-right: 20px;
background: #333333;
border: none;
margin: 0.25rem;
border-radius: 32px;
outline: none;
color: #fff;
cursor: pointer;
filter: contrast(150%);
outline: 2px solid black;
}
#form>button:hover {
opacity: 1;
filter: contrast(100%);
}
.speechbtn:hover {
opacity: 0.8;
}
.speechbtn {
text-align: center;
font-size: large;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

padding-left: 7px;
padding-right: 7px;
opacity: 1;
background-color: #ededed;
outline: 2px solid red;
margin: 0.25rem;
border-radius: 32px;
border: none;
cursor: pointer;
flex-grow: 1;
animation-name: recording;
animation-duration: 2s;
animation-play-state: paused;
animation-iteration-count: infinite;
}
@keyframes recording {
  0% {
    background-color: #ededed;
    outline: 2px solid red;
    filter: contrast(150%);
  }
  25% {
    background-color: red;
    outline: 2px solid #ffffff;
    filter: contrast(130%);
  }
  50% {
    background-color: #ffffff;
    outline: 2px solid red;
    filter: contrast(130%);
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

75% {
    background-color: red;
    outline: 2px solid #ffffff;
    filter: contrast(130%);
}
100% {
    background-color: #ffffff;
    outline: 2px solid red;
    filter: contrast(130%);
}
}
</style>
</head>
<body>
<!-- Open pop up chat -->
<div class="open-button" id="greeting">
<button class="btn" onclick="openForm()">
<h3>Ask Me Question!</h3>

</button>
</div>
<!------- Chatting form ----->
<div class="chat-popup" id="myForm">
<!-- Exit popup chat button-->
<div class="header">
<button class="how2use-btn">วิธีใช้งาน</button>
<button type="button" class="cancel-btn" onclick="closeForm()">
<b style="font-size:18px"> ✖ </b>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    </button>
</div>
<!-- Show output -->
<div class="display-show-output">
    <div class="show-audio">
        <audio autoplay id="audioPlayer">
            <source src="" type="audio/mpeg">
            Your browser does not support the audio element.
        </audio>
    </div>
    <div class="show-output">
        <span id="APlanswer"></span>
    </div>
</div>
<!-- Show input -->
<div class="showTelecomchan">
    <div class="telecomchan"></div>
    <div id="result" class="show-input"></div>
</div>
<!-- input field -->
<div class="footer">
    <form id="form">
        <input id="input">
        <button>Send</button>
    </form>
    <button class="speechbtn" id="speech"></button>
</div>
</div>
<script src="/socket.io/socket.io.js"></script>
<script>
    //----- Open pop-chat -----

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

function openForm() {
  init_msg.innerHTML = "สวัสดีค่ะ มีอะไรจะถามน้องเทเลคอมจิ้งมัยคะ";
  document.getElementById('APlanswer').appendChild(init_msg);
  document.getElementById("myForm").style.display = "block";
  show_output.style.marginTop = '15%';
  show_output.style.marginLeft = '39%';
  show_output.style.width = 'fit-content';
  show_output.style.height = 'fit-content';
  show_output.style.borderRadius = '35px 30px 30px 0px';
  show_input.style.display = 'none';
  telecomchan.style.marginLeft = '2%';
}
function closeForm() {
  document.getElementById("myForm").style.display = "none";
  req_msg = null;
  text = "";
}
//----- initial speech recognition -----
const record = document.createElement("p");
record.innerHTML = "●";
document.getElementById("speech").appendChild(record);
window.addEventListener("DOMContentLoaded", () => {
  const button = document.getElementById("speech");
  const speechbtn = document.querySelector(".speechbtn");
  const result = document.getElementById("result");
  let listening = false;
  const SpeechRecognition = window.SpeechRecognition ||
window.webkitSpeechRecognition;
  if (typeof SpeechRecognition !== "undefined") {
    let req_msg = "";
    const recognition = new SpeechRecognition();

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

recognition.lang = 'th-TH';
const stop = () => {
  recognition.stop();
  button.textContent = "🔴";
  emit(req_msg)
  //console.log(`req_msg: ${req_msg}`)
  speechbtn.style.animationPlayState = 'initial';
  speechbtn.style.animationIterationCount = '0';
};
const start = () => {
  recognition.start();
  button.textContent = "🔴";
  result.style.display = "none";
  speechbtn.style.animationPlayState = 'running';
  speechbtn.style.animationIterationCount = 'infinite';
};
const onResult = event => {
  result.innerHTML = "";
  for (const res of event.results) {
    req_msg = res[0].transcript;
    const text = document.createTextNode(res[0].transcript);
    const p = document.createElement("p");
    if (res.isFinal) {
      p.classList.add("final");
    }
    p.appendChild(text);
    result.appendChild(p);
  }
  //console.log(result);
  result.style.display = "block";
};

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

recognition.continuous = true;
recognition.interimResults = true;
recognition.addEventListener("result", onResult);
button.addEventListener("click", event => {
    listening ? stop() : start();
    listening = !listening;
});
} else {
    button.remove();
    const message = document.getElementById("speech");
    message.removeAttribute("hidden");
    message.setAttribute("aria-hidden", "false");
}
});
//----- greeting -----
const greeting = document.getElementById('greeting');
const init_msg = document.createElement("p");
const telecomchan = document.querySelector(".telecomchan");
const how2use_btn = document.querySelector('.how2use-btn');
const show_output = document.querySelector('.show-output');
const show_input = document.querySelector('.show-input');
//How to use btn
let btn_event = true;
const btn_1stEvent = event => {
    init_msg.innerHTML = "วิธีการใช้งาน: คุณสามารถถามคำถามจากการพิมพ์หรือพูดก็ได้
    ดังนี้<ul><li>สามารถพิมพ์ถามคำถามและกดปุ่ม Send หรือจะกด Enter ก็ได้</li><li>สามารถ
    คำถามด้วยเสียงได้โดยการกดปุ่ม Record ทางด้านล่างขวามือและกดอีกครั้งเมื่อต้องการหยุด
    </li></ul>ข้อแนะนำ: หากไม่สามารถใช้ไมโครโฟนได้ให้ตรวจการอนุญาตการเข้าถึงไมโครโฟนของ
    Web Browser";
    document.getElementById('APlanswer').appendChild(init_msg);
    show_output.style.marginTop = '12%';

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

show_output.style.borderRadius = '20px';
show_output.style.width = '57%';
show_output.style.height = '60%';
show_output.style.display = 'block';
show_input.style.display = 'none';
show_output.style.marginLeft = '33%';
telecomchan.style.marginLeft = '0%';
}
let text = null;
const btn_nthEvent = event => {
  if (req_msg) {
    show_output.style.marginTop = '15%';
    show_output.style.width = 'fit-content';
    show_output.style.borderRadius = '35px 30px 30px 0px';
    show_output.style.display = 'block'
    show_output.style.height = 'fit-content';
    show_input.style.display = 'block'
    //show_output.style.overflow = 'auto';
    //telecomchan.style.marginLeft = '2%';
    init_msg.innerHTML = text;
    document.getElementById('APlanswer').appendChild(init_msg);
  } else {
    init_msg.innerHTML = "สวัสดีค่ะ มีอะไรจะถามน้องเทเลคอมจิ้งมัยคะ";
    document.getElementById('APlanswer').appendChild(init_msg);
    show_output.style.marginTop = '15%';
    show_output.style.marginLeft = '39%';
    show_output.style.width = 'fit-content';
    show_output.style.height = 'fit-content';
    show_output.style.borderRadius = '35px 30px 30px 0px';
    show_input.style.display = 'none'
    telecomchan.style.marginLeft = '2%';
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
  }
  how2use_btn.addEventListener('click', async event => {
    btn_event ? await btn_1stEvent() : await btn_nthEvent();
    btn_event = !btn_event
  });
  //----- submit form -----
  let form = document.getElementById('form');
  let input = document.getElementById('input');
  let req_msg = null
  form.addEventListener('submit', function (e) {
    e.preventDefault();
    if (input.value === "") {
      if (req_msg !== null) {
        document.getElementById("result").style.display = "block";
      } else {
        document.getElementById("result").style.display = "none";
      }
    } else {
      document.getElementById("result").style.display = "block";
      //send MSG to server
      req_msg = input.value
      emit(req_msg)
      document.getElementById("result").innerHTML = req_msg;
      input.value = ""
    }
  });
  //----- socket.io -----
  const client = io();
  //----- emit data to server -----
  function emit(input) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

if (input !== "") {
  client.emit('messages', input);
  input.value = "";
}
//console.log(`ClientID: <${(client.id)}> Request: `, input);
}
client.on('new-message', (res) => {
  text = res;
  init_msg.innerHTML = text;
  document.getElementById('APlanswer').appendChild(init_msg);
});
//greeting
async function greetingfromserver() {
  const data = "สวัสดีค่ะ มีอะไรจะถามน้องเทเลคอมจิ้งมัยคะ"
  init_msg.innerHTML = data;
  document.getElementById('APlanswer').appendChild(init_msg);
  telecomchan.style.backgroundImage =
"url('https://res.cloudinary.com/dgumpoqcu/image/upload/v1678897044/hi1_0000_1_l
ukgiv.gif')";
  await client.emit('greetingtext', data);
  await client.on('greeting', (audioContent) => {
    console.log('Greeting Audio:')
    console.log(audioContent)
    greeting.addEventListener('click', async () => {
      const arrayBuffer = new Blob([audioContent]);
      const audio = document.getElementById('audioPlayer');
      audio.src = URL.createObjectURL(arrayBuffer);
      audio.style.display = 'none';
      const playPromise = audioPlayer.play();
      if (playPromise !== undefined) {
        playPromise.then(function () { // Automatic playback started!

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const myaudio = document.getElementsByTagName("audio")[0]
const durationTime = myaudio.duration.toFixed(0)
//console.log(durationTime)
telecomchan.style.backgroundImage =
"url('https://res.cloudinary.com/dgumpoqcu/image/upload/v1678897044/hi1_0000_1_l
ukgiv.gif)";

const timerId = setInterval(function () {
    telecomchan.style.backgroundImage =
"url('https://res.cloudinary.com/dgumpoqcu/image/upload/v1678897044/hi1_0000_1_l
ukgiv.gif)";

    if (Number(myaudio.currentTime.toFixed(0)) ===
(Number(durationTime) - 1)) {
        //console.log("loop done...", myaudio.currentTime)
        telecomchan.style.backgroundImage =
"url('https://res.cloudinary.com/dgumpoqcu/image/upload/v1678897045/Default_1_00
00_ynugdd.gif)";
        clearInterval(timerId);
    }
}, 1000)
}).catch(function (error) { /*Automatic playback failed.* / });
});
});
}
greetingfromserver()
//answering
client.on('speech-audio', async (audioContent) => {
    console.log('Answer Audio:')
    console.log(audioContent);
    const arrayBuffer = new Blob([audioContent]);
    const audio = document.getElementById('audioPlayer');

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

audio.src = URL.createObjectURL(arrayBuffer);
audio.style.display = 'none';
const playPromise = audioPlayer.play();
if (playPromise !== undefined) {
  playPromise.then(function () {
    const myaudio = document.getElementsByTagName("audio")[0]
    const durationTime = myaudio.duration.toFixed(0)
    telecomchan.style.backgroundImage =
"url('https://res.cloudinary.com/dgumpoqcu/image/upload/v1678897044/answer_1_00_00_frt1lh.gif')"
    const timerId = setInterval(function () {
      telecomchan.style.backgroundImage =
"url('https://res.cloudinary.com/dgumpoqcu/image/upload/v1678897044/answer_1_00_00_frt1lh.gif')"
      if (myaudio.currentTime.toFixed(0) === durationTime) {
        telecomchan.style.backgroundImage =
"url('https://res.cloudinary.com/dgumpoqcu/image/upload/v1678897045/Default_1_00_00_ynugdd.gif')"
        clearInterval(timerId);
      }
    }, 1000)
  }).catch(function (error) {
    // Automatic playback failed.
  });
}
});
</script>
</body>
</html>

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

require('dotenv').config();
const express = require('express');
const app = express();
const http = require('http');
const server = http.createServer(app);
const { Server } = require("socket.io");
const io = new Server(server);
const fs = require('fs');
const path = require('path');
const pub = path.join(__dirname, 'Public');
const PORT = process.env.PORT || 5000;
const textToSpeech = require('@google-cloud/text-to-speech');
const client = new textToSpeech.TextToSpeechClient();
app.get('/', (req, res) => {
  //res.sendFile(path.join(pub, 'index.html'))
  const stream = fs.createReadStream(path.join(pub, 'index.html'));
  stream.pipe(res);
  io.on('connection', async (socket) => {
    console.log("")
    console.log('-----')
    console.log(`a user connected: ${socket.id}`);
    console.log(`Client ip address: ` + socket.handshake.address);
    console.log(`Date-time: ` + socket.handshake.time);
    console.log('-----')
    console.log("")
    socket.on('disconnect', () => { //client disconnected
      console.log(`user disconnected: ${socket.id}`);
    });
    //for TTS greeting
    async function synthesizeSpeech(text) {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

const request = {
  input: { text },
  voice: { languageCode: 'th-TH', ssmlGender: 'NEUTRAL' },
  audioConfig: {
    audioEncoding: 'MP3',
    pitch: 6,
    speakingRate: 1.1
  },
},
};
const [response] = await client.synthesizeSpeech(request);
const audioContent = response.audioContent;
socket.emit('greeting', audioContent);
}
socket.on('greetingtext', (text) => {
  synthesizeSpeech(text);
})
const url = 'https://telecom-qa-app-oio6zs2epa-as.a.run.app/predict';
const options = {
  method: 'POST',
  headers: { 'content-type': 'application/json', }
};
//req question from client
await socket.on('messages', (msg) => {
  console.log(`User <${(socket.id)}> request =>`, msg);
  options.body = `{"text":"${msg}"}`;
  fetch(url, options)
    .then(res => res.json())
    .then(async json => {
      if (typeof json.answer !== "undefined") {
        const text = json.answer + 'ค่ะ';
        await synthesizeSpeech(text)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

await socket.emit('new-message', (text)); // answer and audio to
client

async function synthesizeSpeech(text) {
  const request = {
    input: { text },
    voice: { languageCode: 'th-TH', ssmlGender: 'FEMALE' },
    audioConfig: {
      audioEncoding: 'MP3',
      pitch: 6,
      speakingRate: 1.05
    },
  };
  const [response] = await client.synthesizeSpeech(request);
  const audioContent = response.audioContent;
  socket.emit('speech-audio', audioContent);
}
} else {
  await console.error('error:' + " Endpoint Crashed");
  await socket.emit('new-message', 'ขอภัยในความไม่สะดวก เนื่องจากอยู่ใน
การปรับปรุงระบบชั่วคราวค่ะ'); // answer to client
  await synthesizeSpeech('ขอภัยในความไม่สะดวก เนื่องจากการ
ปรับปรุงระบบชั่วคราวค่ะ')
  async function synthesizeSpeech(text) {
    const request = {
      input: { text },
      voice: { languageCode: 'th-TH', ssmlGender: 'FEMALE' },
      audioConfig: {
        audioEncoding: 'MP3',
        pitch: 5,
        speakingRate: 1.05
      },
    },

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    };
    const [response] = await client.synthesizeSpeech(request);
    const audioContent = response.audioContent;
    socket.emit('speech-audio', audioContent);
  }
}
})
.catch(err => {
  console.error('error:' + err);
  socket.emit('new-message', 'ขอภัยในความไม่สะดวก เนื่องจากอยู่ในการ
ปรับปรุงระบบชั่วคราวค่ะ'); // answer to client
  synthesizeSpeech('ขอภัยในความไม่สะดวก เนื่องจากอยู่ในการปรับปรุงระบบ
ชั่วคราวค่ะ')
  async function synthesizeSpeech(text) {
    const request = {
      input: { text },
      voice: { languageCode: 'th-TH', ssmlGender: 'FEMALE' },
      audioConfig: {
        audioEncoding: 'MP3',
        pitch: 5,
        speakingRate: 1.05
      },
    };
  };
  const [response] = await client.synthesizeSpeech(request);
  const audioContent = response.audioContent;
  socket.emit('speech-audio', audioContent);
}
});
});
});
});

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
server.listen(PORT, () => {  
  console.log(`Express Server Listening on ${PORT}`);  
});
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้