**ห้องสอบ ECC-802**

**CE62-22**

**ประเภท SW\_Dev**

**การตรวจสอบความถูกต้องของหนังสือเสียงโดยใช้**

**Machine Learning และ AI**

**ASSESSING ACCURACY OF AUDIOBOOKS BY USING**

**MACHINE LEARNING AND AI**

**ภัคพงษ์ ปุรสาชิต**

**รณกฤต วงษ์อินทร์**

**รายงานนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต**

**สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์**

**สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง**

**ปีการศึกษา 2563**

ปริญญานิพนธ์ปีการศึกษา 2563 ภาควิชาวิศวกรรมคอมพิวเตอร์  
คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การตรวจสอบความถูกต้องของหนังสือเสียงโดยใช้ Machine Learning และ AI

ASSESSING ACCURACY OF AUDIOBOOKS BY USING

MACHINE LEARNING AND AI

ผู้จัดทำ  
1. นายภัคพงษ์ ปุรสาชิต 60010760  
2. นายรณกฤต วงษ์อินทร์ 60010843

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ อาจารย์ที่ปรึกษา   
 (ผศ.อัครเดช วัชระภูพงษ์)

**การตรวจสอบความถูกต้องของหนังสือเสียงโดยใช้**

**Machine Learning และ AI**

นายภัคพงษ์ ปุรสาชิต 60010760

นายรณกฤต วงษ์อินทร์ 60010814

ผศ.อัครเดช วัชระภูพงษ์ อาจารย์ที่ปรึกษา

ปีการศึกษา 2563

**บทคัดย่อ**

โครงงานฉบับนี้นำเสนอการออกแบบและพัฒนาระบบการตรวจสอบความถูกต้องของหนังสือเสียงโดยใช้ Machine Learning และ AI โดยการศึกษาในด้าน Speech Recognition เพื่อนำมาพัฒนาโปรแกรมโดยตัวซอฟต์แวร์จะใช้ภาษา Python ในการพัฒนา AI และ Machine Learning บน Google Cloud Platform เพื่อให้เกิดประโยชน์แก่เป้าหมายที่เป็นกลุ่มเด็กซึ่งมีความบกพร่องทางสายตา

โครงงานนี้จะเป็นเครื่องมือซึ่งช่วยเสริมสร้างพัฒนาการด้านการฟังและการพูดออกเสียงภาษาไทย โดยมีเกณฑ์การตรวจสอบและวัดผลเพื่อให้ได้ผลลัพธ์ซึ่งสอดคล้องไปกับทฤษฎีและผลงานวิจัยที่เกี่ยวข้องตั้งแต่ด้านการสะกดคำที่ถูกต้อง ความเร็วในการพูด จนไปถึงการวรรคตอนที่เหมาะสมแก่การเรียนรู้สำหรับกลุ่มเป้าหมาย

**ASSESSING ACCURACY OF AUDIOBOOKS BY USING**

**MACHINE LEARNING AND AI**

Pakkapong Purasachit 60010760

Ronnakrit Wongin 60010814

Asst.Prof.Akkradach Watcharapupong Advisor

Academic Year 2020

**ABSTRACT**

This project presents designs and development of Audiobooks accuracy assessing by Machine Learning and AI with research about Speech Recognition for software development. The Software uses Python to develop AI and Machine Learning on Google Cloud Platform for benefits to visually impaired children.

This Project will be a tool to develop listening skills in Thai and also have testing measurement to conform the results to the related theories and researches including spelling correction, speaking speed, and properly length of space to the target group of children.

สารบัญ

บทคัดย่อภาษาไทย...............................................................................................................ก

บทคัดย่อภาษาอังกฤษ..........................................................................................................ข

คำนำ....................................................................................................................................ค

สารบัญ.................................................................................................................................ง

สารบัญรูป.............................................................................................................................จ

สารบัญตาราง.......................................................................................................................ฉ

[บทที่ 1 9](#_Toc56459676)

[1.1 ความเป็นมาของปัญหา 9](#_Toc56459677)

[1.2 วัตถุประสงค์ของการศึกษา 10](#_Toc56459678)

[1.3 เป้าหมาย 10](#_Toc56459679)

[1.4 ขอบเขตของโครงงาน 10](#_Toc56459680)

[1.5 ขั้นตอนการดำเนินงาน 10](#_Toc56459681)

[บทที่ 2 12](#_Toc56459682)

[2.1 Speech Recognition 12](#_Toc56459683)

[2.2 API 13](#_Toc56459684)

[2.3 gRPC 15](#_Toc56459685)

[2.4 RESTful หรือ REST 15](#_Toc56459686)

[2.5 Google API Speech to Text 16](#_Toc56459687)

[2.7 ประเภทของไฟล์เสียง 18](#_Toc56459688)

[2.8 งานวิจัยที่เกี่ยวข้อง (Related Works) 19](#_Toc56459689)

[บทที่ 3 21](#_Toc56459690)

[3.1 การจัดเตรียม server และเครื่องมือที่จำเป็น 21](#_Toc56459691)

[3.2 การจัดหา Dataset สำหรับทำการทดสอบ 26](#_Toc56459692)

[3.3 การสร้างเกณฑ์สำหรับการตรวจสอบคุณภาพของหนังสือเสียง 27](#_Toc56459693)

[3.4 ภาพรวมการทำงานของระบบ 27](#_Toc56459694)

[3.5 การออกแบบการทำงานและโครงสร้างของสถาปัตยกรรม 28](#_Toc56459695)

[บทที่ 4 30](#_Toc56459696)

[4.1 ทดสอบเดโม่ Word Matching Module กับตัวอย่างของ Google 30](#_Toc56459697)

[4.2 ทดสอบเดโม่ Word Matching Module กับตัวอย่างภาษาอังกฤษ 34](#_Toc56459698)

[4.3 ทดสอบเดโม่ Word Matching Module กับตัวอย่างภาษาไทย 35](#_Toc56459699)

[บทที่ 5 37](#_Toc56459700)

[5.1 ข้อสรุป 37](#_Toc56459701)

[5.2 ข้อเสนอแนะ 37](#_Toc56459702)

[5.3 ปัญหาและอุปสรรคที่พบ 37](#_Toc56459703)

[5.4 สิ่งที่จะทำต่อไป 37](#_Toc56459704)

[บรรณานุกรม 39](#_Toc56459705)

สารบัญรูป

บทที่ 2

ภาพที่ 2.1 แผนภาพแสดงการติดต่อของ API .................................................................. 14

ภาพที่ 2.2 รูปภาพแสดงการนำเอา API ไปใช้งาน ........................................................... 15

ภาพที่ 2.3 แผนภาพแสดง RESTFul ................................................................................ 16

บทที่ 3

ภาพที่ 3.1 ภาพแสดงหน้าลงทะเบียน Google cloud platform ......................................... 21

ภาพที่ 3.2 ภาพแสดงหน้าลงทะเบียน Google cloud platform หน้ากรอกรายละเอียด ..... 22

ภาพที่ 3.3 ภาพแสดงหน้าลงทะเบียน Google cloud platform หน้ากรอกข้อมูล payment 22

ภาพที่ 3.4 ภาพแสดงหน้า Home ของ Google cloud platform ......................................... 23

ภาพที่ 3.5 ภาพวิธีการสร้าง VM instances ....................................................................... 23

ภาพที่ 3.6 ภาพวิธีการสร้าง VM instances (2) .................................................................. 24

ภาพที่ 3.7 ภาพวิธีการ setting VM instances .................................................................... 24

ภาพที่ 3.8 ภาพยืนยันการสร้าง VM instances .................................................................. 25

ภาพที่ 3.9 ภาพวิธีการ setting VM instances (2) .............................................................. 25

ภาพที่ 3.10 ภาพแสดงหน้าตัวอย่าง Server ....................................................................... 25

ภาพที่ 3.11 Visual Studio Code ....................................................................................... 26

ภาพที่ 3.12 Switch Sound File Converter ........................................................................ 26

ภาพที่ 3.13 แสดงการทำงานของ Word matching module .............................................. 28

ภาพที่ 3.14 แสดงการทำงานของ Speech recognition module ......................................... 29

บทที่ 4

ภาพที่ 4.1 demo files ........................................................................................................ 30

ภาพที่ 4.2 data.json .......................................................................................................... 31

ภาพที่ 4.3 main.py ........................................................................................................... 31

ภาพที่ 4.4 word\_matching.py .......................................................................................... 32

ภาพที่ 4.5 wordMatcher\_demo ........................................................................................ 32

ภาพที่ 4.6 transcribe\_file 1 .............................................................................................. 33

ภาพที่ 4.7 transcribe\_file 2 .............................................................................................. 33

ภาพที่ 4.8 result 1 ............................................................................................................ 34

ภาพที่ 4.9 result 2 ............................................................................................................ 35

ภาพที่ 4.10 result 3 .......................................................................................................... 36

สารบัญตาราง

บทที่ 1

ตารางที่ 1.1 ตารางแสดงระยะเวลาการดำเนินงาน ........................................................... 12

# บทที่ 1

**บทนำ**

## 1.1 ความเป็นมาของปัญหา

ในปัจจุบันเทคโนโลยีแปลงเสียงพูดเป็นข้อความหรือที่เราเรียกกันว่า Speech Recognition นั้น มีบทบาทอย่างมากต่อการดำรงชีวิตประจำวันของเรา ทั้งใน Smart phone ที่นำเทคโนโลยีนี้เข้ามาเพื่ออำนวยความสะดวกแก่ผู้ใช้งาน เช่น การพิมพ์ข้อความด้วยเสียง ตลอดจนการใช้งานระบบสั่งการด้วยเสียง ในสมาร์ทโฟน หรือ สมาร์ทโฮม เป็นต้น นอกจากนั้นยังมีการใช้เทคโนโลยีนี้บนอุปกรณ์อื่น เช่น ระบบคำสั่งเสียงสำหรับหุ่นยนตร์ เป็นต้น

เนื่องจากในปัจจุบันนี้มีการจัดทำหนังสือเสียงสำหรับผู้บกพร่องทางสายตาและผู้สูงอายุ หรือคนปกติที่ต้องการรับรู้เนื้อหาของหนังสือผ่านทางการฟังได้มีโอกาสรับรู้ข่าวสารข้อมูลจากสื่อสิ่งพิมพ์เพิ่มขึ้น ทั้งในด้านของการศึกษาหรือเพื่อความบันเทิง แต่หนังสือเสียงนั้นก็อาจจะมีความผิดพลาดอยู่บ้าง ไม่ว่าจะเป็นการออกเสียงหรือการสะกดคำที่ทำให้สารที่ได้รับนั้นมีความหมายผิดไป ทางกลุ่มเราจึงเล็งเห็นถึงความสำคัญในการพัฒนาคุณภาพของหนังสือเสียงให้ดียิ่งขึ้นด้วยการตรวจสอบคุณภาพของหนังสือเสียง โดยการนำเทคโนโลยี Speech Recognition ที่แปลงจากเสียงพูดไปเป็นตัวอักษรและนำเอาตัวอักษรเหล่านั้นไปเปรียบเทียบกับหนังสือต้นฉบับ เพื่อวัดความถูกต้องของหนังสือเสียงว่ามีความถูกต้องมากน้อยเพียงใด เพื่อให้เป็นประโยชน์ต่อทั้งผู้ที่ใช้งานหนังสือเสียงตลอดจนอาสาสมัครที่เป็นผู้จัดทำ

โครงงาน **“การตรวจสอบความถูกต้องของหนังสือเสียง โดยใช้ machine learning และ AI” (Assessing Accuracy of Audiobooks by Using Machine Learning and AI)** นี้จึงถูกจัดทำขึ้น เพื่อทำการศึกษาเทคโนโลยี Speech Recognition โดยการนำเทคโนโลยีนี้มาใช้ในการวัดความถูกต้องของหนังสือเสียง (Audiobooks) เพื่อบอกค่าความถูกต้องของหนังสือเสียงเหล่านั้น โดยผลสำเร็จจะเป็นโปรแกรมระบบวัดความถูกต้องของหนังสือเสียงโดยใช้เทคโนโลยี machine learning และ AI

## 1.2 วัตถุประสงค์ของการศึกษา

1.2.1 เพื่อพัฒนาคุณภาพของหนังสือเสียงให้ดียิ่งขึ้น ทั้งในแง่ของความถูกต้อง ความชัดเจน การวรรคตอน ตลอดจนความเร็วที่เหมาะสม

1.2.2 เพื่อให้ผู้ใช้งานโปรแกรมสามารถทราบถึงข้อผิดพลาดและแก้ไขปรับปรุง เพื่อพัฒนาทางด้านการอ่านให้ดีขึ้นได้

1.2.3 มุ้งเน้นเพื่อให้เกิดประโยชน์แก่กลุ่มเป้าหมายที่เป็นเด็กซึ่งมีความบกพร่องทางสายตา เพื่อเป็นหนึ่งในเครื่องมือซึ่งสามารถช่วยเสริมสร้างพัฒนาการทางด้านการฟังและการพูดออกเสียง

## 1.3 เป้าหมาย

เพื่อวัดความถูกต้องของหนังสือเสียงด้วย machine learning และ AI

## 1.4 ขอบเขตของโครงงาน

1.4.1 ตัวโปรแกรมรองรับเฉพาะภาษาไทย

1.4.2 ใช้กลุ่มข้อมูลเฉพาะหนังสือนิทานสำหรับเด็ก

1.4.3 ผู้อ่านจะต้องอ่านออกเสียงไม่เร็วเกินไปและชัดถ้อยชัดคำ

## 1.5 ขั้นตอนการดำเนินงาน

1.5.1 ศึกษาทฤษฎีและผลงานที่เกี่ยวข้อง

1.5.2 พัฒนาส่วนของ Dataset

1.5.3 พัฒนาส่วนของ Word Matching

1.5.4 พัฒนาส่วน Speech Recognition

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ขั้นตอนการดำเนินงาน | พ.ศ.2563 | | | | | | | | | | | | | | | | | | | | พ.ศ.2564 | | | |
| สิงหาคม | | | | กันยายน | | | | ตุลาคม | | | | พฤศจิกายน | | | | ธันวาคม | | | | มกราคม | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| **ศึกษาทฤษฎีและผลงานทางวิชาการที่เกี่ยวข้อง** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 1.ศึกษาทฤษฎีและผลงานทางวิชาการที่เกี่ยวข้องกับการรับรู้ของเด็กตาบอด |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2.ศึกษาทฤษฎีและผลงานทางวิชาการที่เกี่ยวกับ Speech Recognition API |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3.ศึกษาทฤษฎีและผลงานทางวิชาการที่เกี่ยวกับการสอนเด็กปฐมวัย |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ส่วน Speech Recognition** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4.ศึกษาการใช้งาน Speech Recognition API |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5.ศึกษาและพัฒนาส่วน File Handling |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6.พัฒนาส่วน Audiobooks to text |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7.พัฒนาส่วนตรวจสอบน้ำเสียงการอ่าน |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8.ทดสอบและแก้ไขส่วน Speech Recognition |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ส่วน Dataset** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9.จัดหาและพัฒนา dataset สำหรับใช้ทดสอบ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10.จัดหาและพัฒนา dataset |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **ส่วน Word matching** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11.ศึกษาเพื่อออกแบบเกณฑ์การวัดค่าความถูกต้องของหนังสือเสียง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12.ออกแบบเกณฑ์การวัดค่าความถูกต้องของหนังสือเสียง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13.ศึกษาเกี่ยวกับ Machine Learning และ AI |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14.ศึกษาเกี่ยวกับการเขียน code เพื่อเรียกใช้ API ต่างๆ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15.พัฒนาส่วน Word matching |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16.พัฒนาส่วนตรวจสอบความเร็วในการอ่าน |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 17.พัฒนาส่วนตรวจสอบวรรคตอนในการอ่าน |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 18.ทดสอบและแก้ไขส่วน Word matching |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **รวมทั้ง 2 Module เข้าด้วยกัน** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 19.ออกแบบระบบวัดความถูกต้องของหนังสือเสียง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 20.พัฒนาระบบวัดความถูกต้องของหนังสือเสียง |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

ตาราง 1.1 ตารางแสดงระยะเวลาการดำเนินงาน

# บทที่ 2

**ทฤษฎีและหลักการที่เกี่ยวข้อง**

## 2.1 Speech Recognition

กระบวนการ Speech Recognition คือ กระบวนการที่เอาเสียงพูดของมนุษย์นำไปตีความและถอดความให้กลายเป็นตัวอักษร โดยวิธีการแบ่งเสียงพูดของมนุษย์ออกเป็นส่วนๆ และทำการวิเคราะห์เสียงแต่ละส่วน โดยการใช้อัลกอริทึมเพื่อหาคำที่เหมาะสมที่สุดกับภาษานั้นและทำการแปลงภาษาเหล่านั้นเป็นตัวอักษร โดยอาศัยกระบวนการดังนี้

การนำเสียงพูดที่บันทึกลงใน neural network ไปแปลงเสียงพูดที่ถูกบันทึกเป็นข้อความ

กระบวนการดังกล่าวมีปัญหาคือ ความเร็วในการออกเสียงพูดของแต่ละคนไม่เท่ากัน ซึ่งทำให้ขนาดของข้อมูลที่ถูกเก็บมีขนาดต่างกัน ซึ่งตามหลักการแล้ว ไฟล์เสียงทั้งสองควรต้องจัดการให้เป็นข้อความเดียวกัน ซึ่งการจัดการไฟล์เสียงทั้ง 2 ที่มีความยาวไม่เท่ากัน โดยอัตโนมัติให้เป็นข้อความที่มีความยาวคงที่นั้นทำได้ค่อนข้างยาก

เพื่อแก้ปัญหาดังกล่าวจำเป็นต้องใช้การ processing เพิ่มเติม นอกเหนือจาก neural network กระบวนการที่แก้ไขปัญหามีขั้นตอนดังนี้

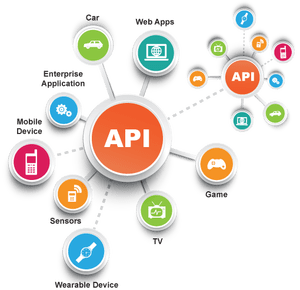
**2.1.1 Turning Sounds into Bits** นำคลื่นเสียงลงไปในคอมพิวเตอร์เพื่อแปลงคลื่นเสียงให้เป็นตัวเลข โดยทำการ sampling และบันทึกตัวเลขที่แสดงถึงแอมพลิจูดของคลื่นเสียงในเวลานั้น

**2.1.2. Pre-processing our Sampled Sound Data** ทำการจัดการกับข้อมูลที่ sampling มาและทำการแบ่งข้อมูลออกเป็นช่วงสั้นๆ แต่เนื่องจากข้อมูลเสียงนั้น มีความซับซ้อน เนื่องจากเป็นเสียงพูดของมนุษย์ซึ่งจะมีความหลากหลายของโทนเสียง จึงต้องใช้เทคนิคในการจำแนกเสียงและทำการ mapped ผลลัพธ์ตามความถี่ของเสียง (ต่ำ - สูง) และนำโมเดลที่ได้เข้าสู่ neural network ทำให้ neural network สามารถค้นหารูปแบบของข้อมูลประเภทนี้ได้ง่ายกว่าคลื่นเสียงแบบดังเดิม

**2.1.3. Recognizing Characters from Short Sounds** นำผลข้อมูลที่ได้จาก neural network ไปป้อนให้กับ recurrent neural network (มีหน่วยความจำที่ทำหน้าที่คาดการณ์) เนื่องจากตัวอักษรแต่ละตัวที่จะคาดการณ์ควรมีผลต่อโอกาสของตัวอักษรถัดไปที่จะเกิดขึ้น เช่นหากพูดว่า "HEL" ก็จะมีโอกาสสูงที่คำถัดไปจะพูดว่า "LO" และมีโอกาสน้อยที่จะพูดสิ่งที่ไม่สามารถออกเสียงได้ถัดไป เช่น "WXY" ดังนั้นการมีหน่วยความจำที่เก็บค่าก่อนหน้าจะช่วยให้ neural network ทำการคาดการณ์ที่แม่นยำขึ้น

## 2.2 API

Application Program Interface (API) ซึ่งคือ คำสั่งที่อนุญาตให้ software program สามารถสื่อสารระหว่างกันได้ API เป็นช่องทางสำหรับขอใช้บริการคำสั่ง จาก operation system หรือ แอพพลิเคชัน อื่น ซึ่งมันใช้งานโดยติดตั้ง function และเรียกใช้งานตาม document ที่เขียนไว้



ภาพที่ 2.1 แผนภาพแสดงการติดต่อของ API

**2.2.1 ส่วนประกอบของ API**

API สร้างขึ้นจากส่วนสำคัญ 2 อย่าง คือ

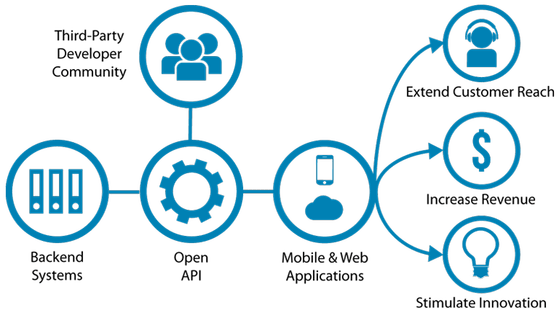
2.2.1.1 ข้อกำหนดที่จะอธิบายการแลกเปลี่ยนข้อมูลระหว่าง program ซึ่งทำออกมาในลักษณะ document เพื่อบอกว่า request/response ต้องเป็นอย่างไร

Software ที่เขียนขึ้นตามข้อกำหนด และทำการเผยแพร่ออกไปให้ใช้งาน

2.2.1.2 โดยปกติแล้ว แอปพลิเคชันที่มี API จะต้องถูกเขียนเป็นภาษา programming และ พัฒนาเพิ่มได้ง่าย จึงจำเป็นต้องมี่การตรวจสอบโครงสร้าง API เพราะฉะนั้น API ที่ดี ผู้ที่ออกแบบต้องให้ความสำคัญในการทดสอบเพื่อตรวจสอบ logic ที่สามารถเกิดขึ้นได้จากการใช้งาน

**2.2.2 การใช้งาน API**

ปัจจุบันนี้ API ถูกใช้งานใน แอปพลิเคชัน เพื่อสื่อสารกับ user โดยไม่จำเป้นต้องมีความรู้ บริษัทใหญ่ๆหลายบริษัทมีการเปิด API ให้ภายนอกเข้ามาใช้งาน เช่น facebook, google, twitter ผู้พัฒนาระบบที่สนใจ สามารถนำเอา API เหล่านี้ไปไปต่อยอด ซึ่งทางบริษัทก็สามารถขยายฐานลูกค้าออกไปได้อีก รูปแบบการนำเอา API ไปใช้งานมีดังนี้



ภาพที่ 2.2 รูปภาพแสดงการนำเอา API ไปใช้งาน

**2.2.2.1 Libraries and frameworks**

API มักจะเอาไปใช้เป็น software library ซึ่งเขียนขึ้นตาม document ในรูปบบภาษา program ที่ต่างกันออกไป ตามความเหมาะสมกับงาน เพื่อเอาไปทำเป็น framework ให้กับระบบใช้ในการสื่อสารหากัน

**2.2.2.2 Operating Systems**

API สามารถใช้งานในการสื่อสารระหว่าง แอปพลิเคชัน และ operating system เช่น POSIX หรือ มาตรฐานการสื่อสารของ OS เองก็มี API เป็น command line เพื่อควบคุมการทำงานของ OS

**2.2.2.3 Remote APIs**

Remote APIs ทำไว้ให้ developer สามารถเข้าควบคุมทรัพยากรผ่านทาง protocol เพื่อให้มีมาตราฐานการสื่อสารเดียวกัน ถึงแม้ว่าจะเป็นคนละ technology เช่น Database API สามารถอนุญาตให้ developer เข้ามาดึงข้อมูลใน database หลากหลายชนิดได้ ผ่าน function เดียวกัน เพราะฉะนั้น remote API จึงถูกใช้บ่อยในงาน maintenance ด้วยทำทำงานที่ฝั่ง client ให้ไปดึงข้อมูลจาก server กลับลงมาทำงาน

**2.2.2.4. Web APIs**

นิยมใช้กันมากในปัจจุบัน เพราะอยู่ในกลุ่มของ HTTP และขยายออกไปสู่รูปแบบ [XML](https://saixiii.com/what-is-xml/) และ [JSON](https://saixiii.com/what-is-json/) ซึ่งโดนรวมแล้วก็คืออยู่บน web service เช่น

SOAP (Simple Object Access Protocol) ใช้ XML format ส่งข้อมูล

REST (Representational State Transfer) สามารถใช้ XML หรือ JSON format ส่งข้อมูล

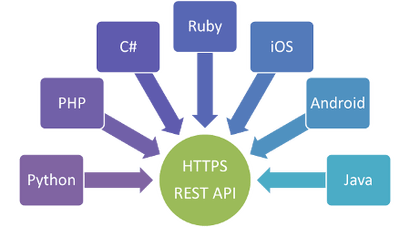
2.3 gRPC

RPC (gRPC Remote Procedure Calls ) เป็นระบบการเรียกใช้ระยะไกลแบบ open source (RPC) ซึ่งพัฒนาครั้งแรกที่ Google ในปี 2558 ใช้ HTTP / 2 สำหรับส่ง Protocol Buffers เป็นภาษาของ Interface และมีความสามารถในการตรวจสอบสิทธิ์สตรีมมิงแบบสองทิศทาง การควบคุมโฟลว์การบล็อก การยกเลิก การกำหนดระยะเวลาสิ้นสุดในกระบวนการ และสร้างการเชื่อมโยงไคลเอนต์ข้ามแพลตฟอร์มและเซิร์ฟเวอร์สำหรับหลายภาษา

## 2.4 RESTful หรือ REST

Representational state transfer หรือ REST คือ การสร้าง Webservice ชนิดหนึ่งที่ใช้สื่อสารกันบน Internet ใช้หลักการแบบ stateless  คือไม่มี session ซึ่งต่างจาก webservice แบบอื่นเช่น [WSDL](https://saixiii.com/what-is-wsdl/)และ [SOAP](https://saixiii.com/what-is-soap/) การทำงานของ RESTful [Webservice](https://saixiii.com/what-is-webservice/) จะอาศัย URI/URL ของ request เพื่อค้นหาและประมวลผลแล้วตอบกลับไปในรูป [XML](https://saixiii.com/what-is-xml/), HTML, [JSON](https://saixiii.com/what-is-json/)  โดย response ที่ตอบกลับจะเป็นการยืนยันผลของคำสั่งที่ส่งมา และสามารถพัฒนาด้วยภาษา programming ได้หลากหลาย คำสั่งก็จะมีตาม HTTP verbs ซึ่งก็คือ

* GET ทำกการดึงข้อมูลภายใน URI ที่กำหนด
* POST สำหรับสร้างข้อมูล
* PUT ใช้แก้ไขข้อมูล
* DELETE สำหรับลบข้อมูล



ภาพที่ 2.3 แผนภาพแสดง RESTFul

**2.4.1 ประวัติของ REST**

REST ถูกตั้งขึ้นโดย Roy Fielding ในปี 2000 ที่ University of California, Irvine ซึ่งได้ทำการพัฒนาขึ้นมาควบคู่กับ HTTP1.1 และ Uniform Resource Identifiers (URI)

**2.4.2 คุณสมบัติของ REST**

เป็น API อย่างหนึ่ง ซึ่งทุกๆ system ต่างใช้ resource ซึ่งเป็นได้ทั้ง image, video, web page หรือข้อมูลทางธรุกิจ ก็ได้ที่สามารถแสดงบนระบบ computer วัตถุประสงค์เพื่อให้ user สามารถเข้าถึง, ติดตั้ง, ปรับแต่ง, ขยาย resource เหล่านี้ได้ง่าย ซึ่งทาง RESTful ได้ออกแบบมาให้มีคุณสมบัติต่อไปนี้

* แสดงผล
* เก็บข้อมูล
* มี URIs
* Stateless ทำงานโดยไม่ต้องมี session
* เชื่อมต่อระหว่าง web service
* Caching

## 2.5 Google API Speech to Text

Speech to Text มี 3 วิธีหลักในการจดจำเสียงพูด คือ

**2.5.1 Speech requests**

1. Synchronous Recognition (REST และ gRPC) ส่งข้อมูลเสียงไปยัง Speech-to-Text API ทำการจดจำข้อมูลนั้นและส่งกลับผลลัพธ์หลังจากประมวลผลเสียงทั้งหมดแล้ว สามารถใช้ได้กับข้อมูลเสียงที่มีความยาว 1 นาทีหรือน้อยกว่า
2. Asynchronous Recognition (REST และ gRPC) ส่งข้อมูลเสียงไปยัง Speech-to-Text API และเริ่มการทำงานระยะยาว เมื่อใช้การดำเนินการนี้สามารถดูผลการจดจำได้เป็นระยะ ๆ โดยใช้ request แบบ Asynchronous สำหรับข้อมูลเสียงในช่วงเวลาใดก็ได้ถึง 480 นาที
3. Streaming Recognition (gRPC เท่านั้น) ทำการจดจำข้อมูลเสียงที่ให้ไว้ภายในสตรีมแบบสองทิศทางของ gRPC request สตรีมมิ่งได้รับการออกแบบมาเพื่อวัตถุประสงค์ในการจดจำแบบเรียลไทม์เช่นการบันทึกเสียงสดจากไมโครโฟน การจดจำการสตรีมให้ผลลัพธ์ระหว่างกาลในขณะที่กำลังบันทึกเสียงทำให้ผลลัพธ์ปรากฏขึ้นเช่นในขณะที่ผู้ใช้ยังคงพูดอยู่

request มีพารามิเตอร์การกำหนดค่าเช่นเดียวกับข้อมูลเสียง ส่วนต่อไปนี้จะอธิบายประเภทของ request การรับรู้คำตอบที่สร้างขึ้นและวิธีจัดการกับคำตอบเหล่านั้นโดยละเอียด

Speech-to-Text API recognition

request recognition Synchronous Speech-to-Text API เป็นวิธีที่ง่ายที่สุดในการดำเนินการจดจำข้อมูลเสียงพูด Speech-to-Text สามารถประมวลผลข้อมูลเสียงพูดได้ถึง 1 นาทีที่ส่งในrequest แบบ Synchronous หลังจากประมวลผล Speech-to-Text และจดจำเสียงทั้งหมดแล้วระบบจะส่งกลับการตอบสนอง

request recognition แบบ Synchronous ถูกบล็อกหมายความว่า Speech-to-Text ต้องส่งคืนการตอบกลับก่อนที่จะประมวลผล request ถัดไป โดยทั่วไป Speech-to-Text จะประมวลผลเสียงได้เร็วกว่าแบบเรียลไทม์โดยจะประมวลผลเสียง 30 วินาทีใน 15 วินาทีโดยเฉลี่ย ในกรณีที่คุณภาพเสียงไม่ดี request อาจใช้เวลานานกว่ามาก

Speech-to-Text มีทั้งวิธี REST และ gRPC สำหรับการเรียกใช้ Speech-to-Text API แบบ Synchronous และ Asynchronous บทความนี้สาธิต REST API เนื่องจากแสดงและอธิบายการใช้งานพื้นฐานของ API ได้ง่ายกว่า

request API คำพูดเป็นข้อความแบบ Synchronous ประกอบด้วยการกำหนดค่าการรู้จำเสียงและข้อมูลเสียง request ตัวอย่างแสดงอยู่ด้านล่าง:

{  
    "config": {  
        "encoding": "LINEAR16",  
        "sampleRateHertz": 16000,  
        "languageCode": "en-US",  
    },  
    "audio": {  
        "uri": "gs://bucket-name/path\_to\_audio\_file"  
    }  
}

request การรู้จำแบบ Synchronous Speech-to-Text API ทั้งหมดต้องมีฟิลด์กำหนดค่าการรู้จำเสียง (ประเภท RecognitionConfig) RecognitionConfig มีฟิลด์ย่อยต่อไปนี้:

* encoding (จำเป็น) ระบุรูปแบบการเข้ารหัสของเสียงที่ให้มา (ประเภท AudioEncoding) หากมีทางเลือกในตัวแปลงสัญญาณให้เลือกเข้ารหัสแบบไม่สูญเสีย เช่น FLAC หรือ LINEAR16 เพื่อประสิทธิภาพที่ดีที่สุด ฟิลด์การเข้ารหัสเป็นทางเลือกสำหรับไฟล์ FLAC และ WAV ที่การเข้ารหัสรวมอยู่ในส่วน file header
* sampleRateHertz (จำเป็น) ระบุ samplerate (Hz) ของเสียงที่ให้มา ฟิลด์ sampleRateHertz เป็นทางเลือกสำหรับไฟล์ FLAC และ WAV ซึ่ง samplerate จะรวมอยู่ในส่วน file header
* languageCode (จำเป็น) มีภาษา + ภูมิภาค / สถานที่ที่จะใช้สำหรับการรู้จำเสียงของเสียงที่ให้มา รหัสภาษาต้องเป็นตัวระบุ BCP-47
* maxAlternatives (ไม่บังคับค่าเริ่มต้นคือ 1) ระบุจำนวนการถอดเสียงทางเลือกที่จะให้ในการตอบกลับ ตามค่าเริ่มต้น Speech-to-Text API จะมีการถอดเสียงหลักหนึ่งรายการ

## 2.7 ประเภทของไฟล์เสียง

โดยทั่วไปแล้วประเภทของไฟล์เสียงแบ่งออกเป็น 2 กลุ่มใหญ่ๆ ได้แก่ LOSSY Audio และ LOSSLESS Audio

**2.7.1 LOSSY Audio**

Lossy Audio คือไฟล์ที่มีการบีบอัดแล้วเสียข้อมูลไปบ้าง ยิ่งบีบอัดให้จำนวน Bitrate น้อยลงเท่าไหร่ คุณภาพเสียงก็จะลดทอนลงไปเท่านั้น เสียงเพลงจากที่ได้ยินรายละเอียดครบถ้วนก็อาจจะลดน้อยถอยลงไปตามลำดับครับ กลุ่มนี้จะค่อนข้างพบเห็นได้ทั่วไป ได้แก่ ไฟล์ประเภท MP3, AAC, OGG, WMA เป็นต้น

**2.7.2 LOSSLESS Audio**

สำหรับ Lossless Audio แปลตามตรงก็หมายถึง ไฟล์ที่ปราศจากการสูญเสียนั่นเอง ซึ่งจุดเด่นของไฟล์ประเภทนี้อยู่ที่คุณภาพของเสียงที่ค่อนข้างทัดเทียมกับแผ่น CD เลยก็ว่าได้ จะสังเกตได้ว่าขนาดของไฟล์ค่อนข้างใหญ่ กลุ่มนี้จะตามหากันยากหน่อย แต่หากพบไฟล์นามสกุลจำพวกนี้รับรองได้ในระดับหนึ่งเลยว่าเป็น Lossless แน่นอน ได้แก่ FLAC, M4A, Wav, Cda, THM, Aiff, APE, TTA, WavPack เป็นต้น

## 2.8 งานวิจัยที่เกี่ยวข้อง (Related Works)

**2.8.1 A Study on Automatic Speech Recognition**

งานวิจัยเกี่ยวกับการอธิบายการสื่อสารกันระหว่างคอมพิวเตอร์กับมนุษย์ที่เรียกว่า Human Computer Interaction (HCI) โดยการใช้เครื่องมือและภาพรวมของคำจำกัดความหลักของ Automatic Speech Recognition (ASR) ซึ่งเป็นส่วนหลักที่สำคัญของ Artificial Intelligence (AI) และงานวิจัยนี้ยังให้บทสรุปของงานวิจัยที่สำคัญที่เกี่ยวข้องกับ speech processing

**2.8.2 Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)**

งานวิจัยเกี่ยวกับการออกแบบเครื่องมือที่จะทำการทดสอบและเปรียบเทียบบริการ speech recognition ของแต่ละบริการ เช่น Microsoft Speech API, Google Speech API และ open-source speech recognition เช่น Sphinx-4 งานวิจัยค้นพบวิธีการที่ดีที่สุดในการเปรียบเทียบ automatic speech recognition systems ในแต่ละบริการ โดยการใช้เสียงที่บันทึก โดยเลือกจากแหล่งที่มาที่แตกต่างกันและทำการคำนวณ word error (WER) ถึงแม้ว่าค่า WER ทั้ง 3 บริการดังกล่าวนั้นจะเป็นที่ยอมรับ แต่พบว่า Google API นั้นเป็นบริการที่ดีที่สุด

**2.8.3 Communication Skills for Children Who Are Blind or Visually Impaired**

เป็นบทความเกี่ยวกับทักษะในการสื่อสารและการพัฒนาทักษะในการสื่อสารของเด็กตาบอดหรือเด็กที่มีความพิการทางสายตาโดยจะกล่าวถึงการสื่อสารในแบบต่างๆได้แก่ การรับรู้ การแสดงออก การสื่อสาร และการพูด ซึ่งทักษะเหล่านี้เป็นทักษะที่เด็กทุกคนจะมีเหมือนกันไม่ว่าจะตาบอด หูหนวกหรือมีความพิการทางด้านต่างๆอย่างใด เด็กบางกลุ่มอาจจะมีการพัฒนาทักษะบางอย่างช้ากว่ากลุ่มอื่นอย่างเช่น การรับรู้ เด็กที่พิการทางสายตาจะไม่สามารถหรือมีความสามารถรับรู้ด้วยการมองเห็นได้น้อยกว่าเด็กที่ตาปกติ จึงจะมีการพัฒนาทักษะการรับรู้ด้วยการฟังเสียงได้มากขึ้นและรองลงมาด้วยการสัมผัส เป็นต้น

**2.8.4 Repeated Interactive Read Alouds in Preschool and Kindergarten**

เป็นบทความที่อ้างอิงถึงงานวิจัยเกี่ยวกับการศึกษาเทคนิคในการอ่านออกเสียงของเด็กปฐมวัยในช่วงก่อนอนุบาลหรือช่วงอนุบาลหรือในช่างอายุ 3-6 ขวบที่ได้ผลดีที่สุด ซึ่งได้ข้อสรุปว่าการอ่านออกเสียงที่ดีที่สุดคือการอ่านออกเสียงโต้ตอบซ้ำๆ (Repeated Interactive Read Alouds) ที่มีความเกี่ยวข้องกับการถามและตอบคำถามและการให้เด็กคาดคะเนเหตุการณ์ล่วงหน้า การที่ให้เด็กอ่านออกเสียงน้อยนั้นจะส่งผลต่อการพัฒนาทักษะการพูด การจดจำคำศัพท์ การรับรู้โดยการฟัง การอ่านที่ให้เด็กได้มีโอกาสโต้ตอบและคาดคะเนเหตุการณ์ล่วงหน้าหรือมีการอธิบายแรงผลักดันของตัวละครหรือเหตุหารณ์ต่างๆจะช่วยพัฒนาทักษะการคิดวิเคราะห์ของเด็กในการให้ความคิดเห็นและการตั้งคำถามได้

# บทที่ 3

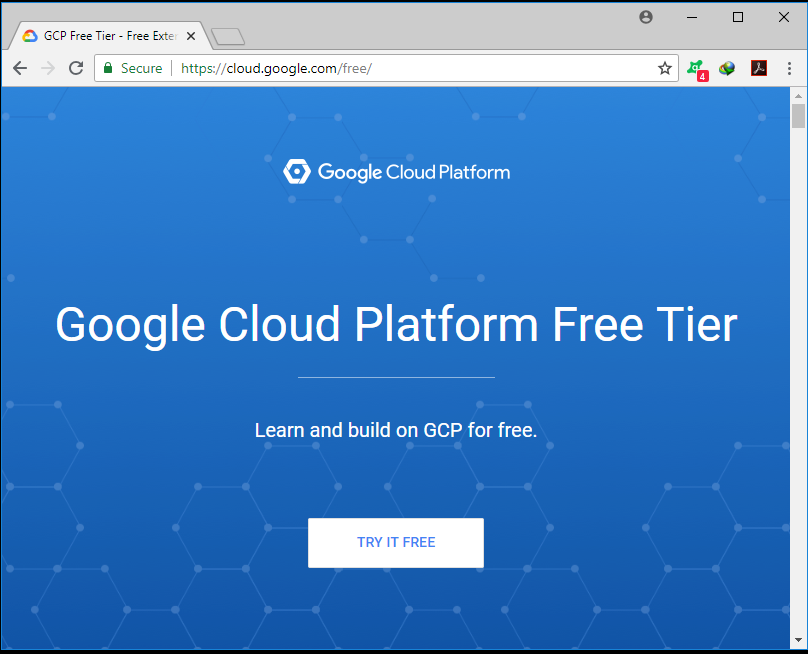
**วิธีการดำเนินงาน**

## 3.1 การจัดเตรียม server และเครื่องมือที่จำเป็น

**3.1.1 รายละเอียดของ server ที่ใช้**

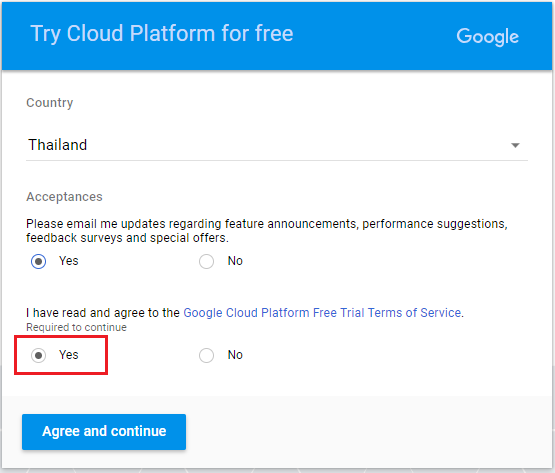
3.1.1.1 Google Cloud Platform

สำหรับการลงทะเบียนใช้งาน Google Cloud Platform (GCP) ครั้งแรก เพื่อรับเครดิตฟรี $300 มีอายุ 12 เดือน ให้เข้าไปที่ <https://cloud.google.com/free/> แล้วคลิก TRY IT FREE



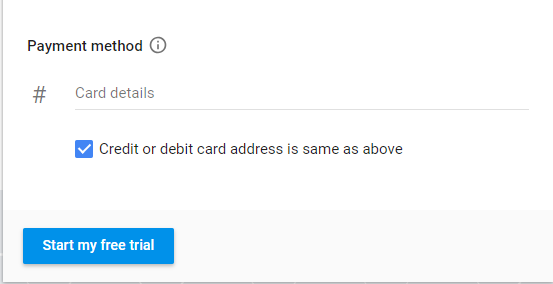
ภาพที่ 3.1 ภาพแสดงหน้าลงทะเบียน Google cloud platform

เลือก Yes และคลิก Agree and continue



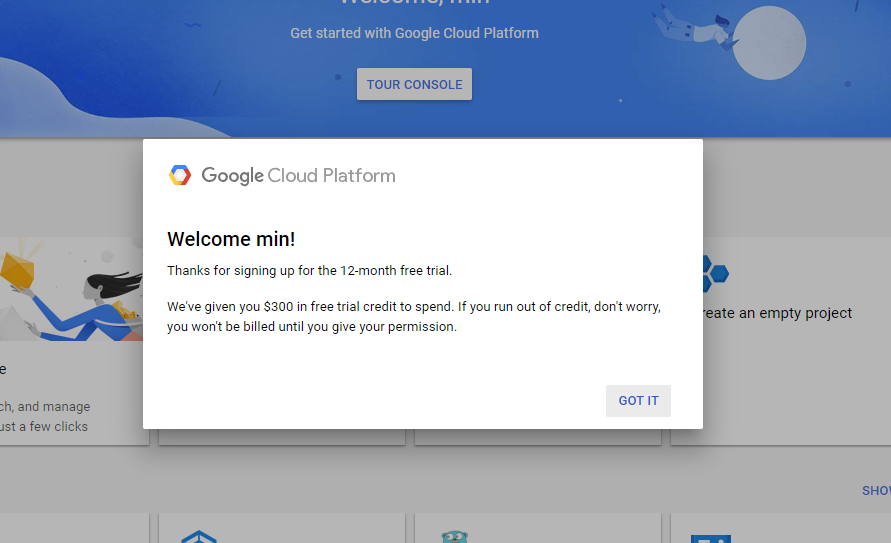
ภาพที่ 3.2 ภาพแสดงหน้าลงทะเบียน Google cloud platform หน้ากรอกรายละเอียด

กรอกรายละเอียดต่างๆ



ภาพที่ 3.3 ภาพแสดงหน้าลงทะเบียน Google cloud platform หน้ากรอกข้อมูล payment

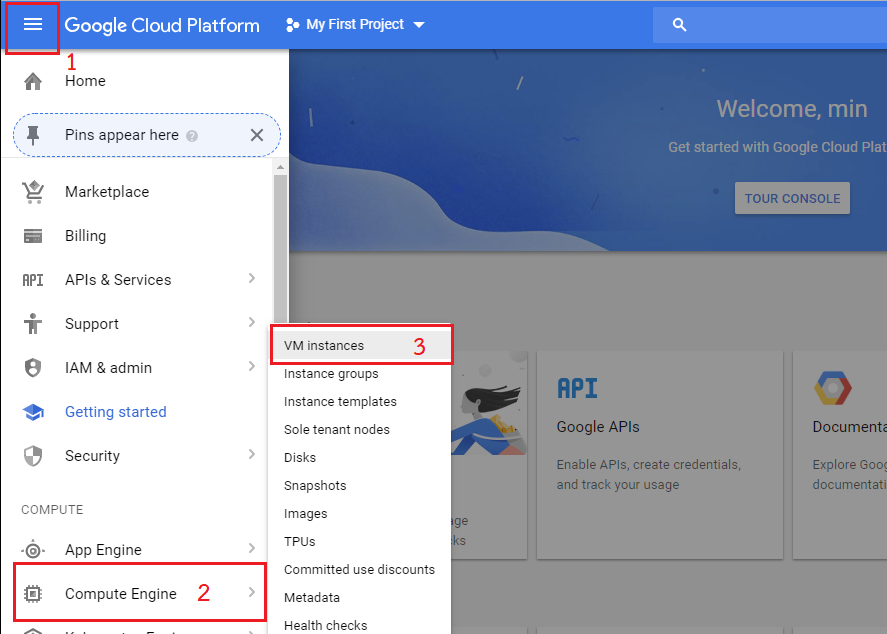
เมื่อลงทะเบียนจะได้รับ $300 ซึ่งมีอายุ 12 เดือน



ภาพที่ 3.4 ภาพแสดงหน้า Home ของ Google cloud platform

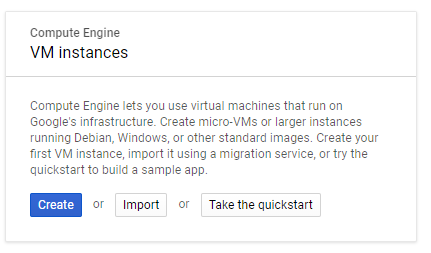
สำหรับการสร้าง instance ให้ทำดังนี้

* คลิก ≡ ที่มุมซ้ายด้านบน
* เลื่อนเมาส์มาที่ Compute Engine
* คลิกที่ VM instances



ภาพที่ 3.5 ภาพวิธีการสร้าง VM instances

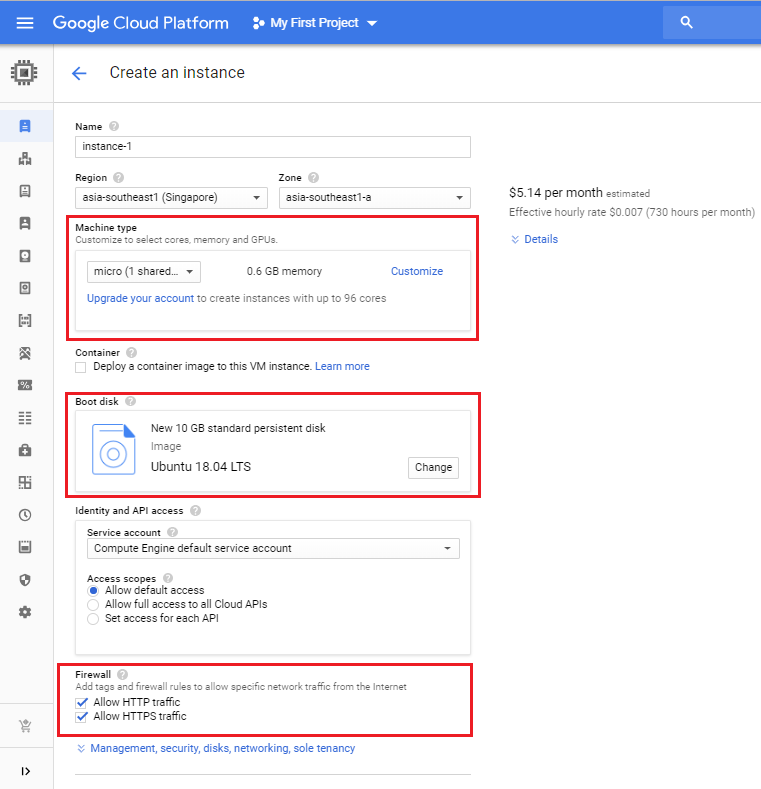
จากภาพต่อมาให้คลิก Create



ภาพที่ 3.6 ภาพวิธีการสร้าง VM instances (2)

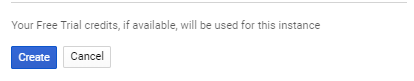
ตั้งค่าได้ตามต้องการโดยจะมีส่วนที่สำคัญคือ

* Machine type จะมีผลกับการเสียเครดิต ตามขนาดที่เลือก
* Boot disk เลือกระบบปฏิบัติการที่ต้องการ



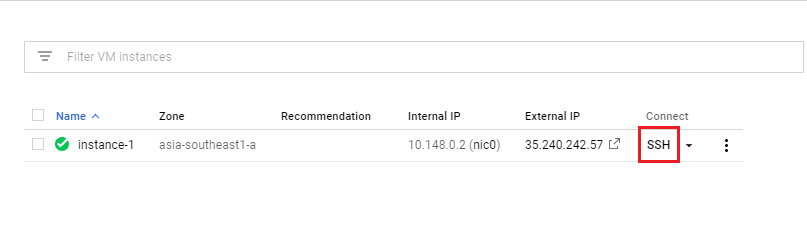
ภาพที่ 3.7 ภาพวิธีการ setting VM instances

เมื่อตั้งค่าได้ตามที่ต้องการแล้ว คลิก Create

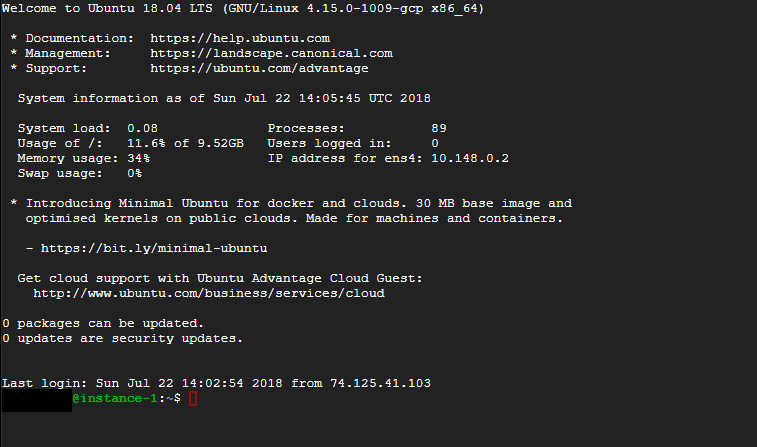


ภาพที่ 3.8 ภาพยืนยันการสร้าง VM instances

หลังจากที่ Create แล้วก็จะได้หน้าตาดังรูป หลังจากนั้นให้คลิกที่ SSH 2 ครั้ง (กดครั้งแรก POP UP จะ Block)

 ภาพที่ 3.9 ภาพวิธีการ setting VM instances (2)

จะพบกับหน้าตา Server

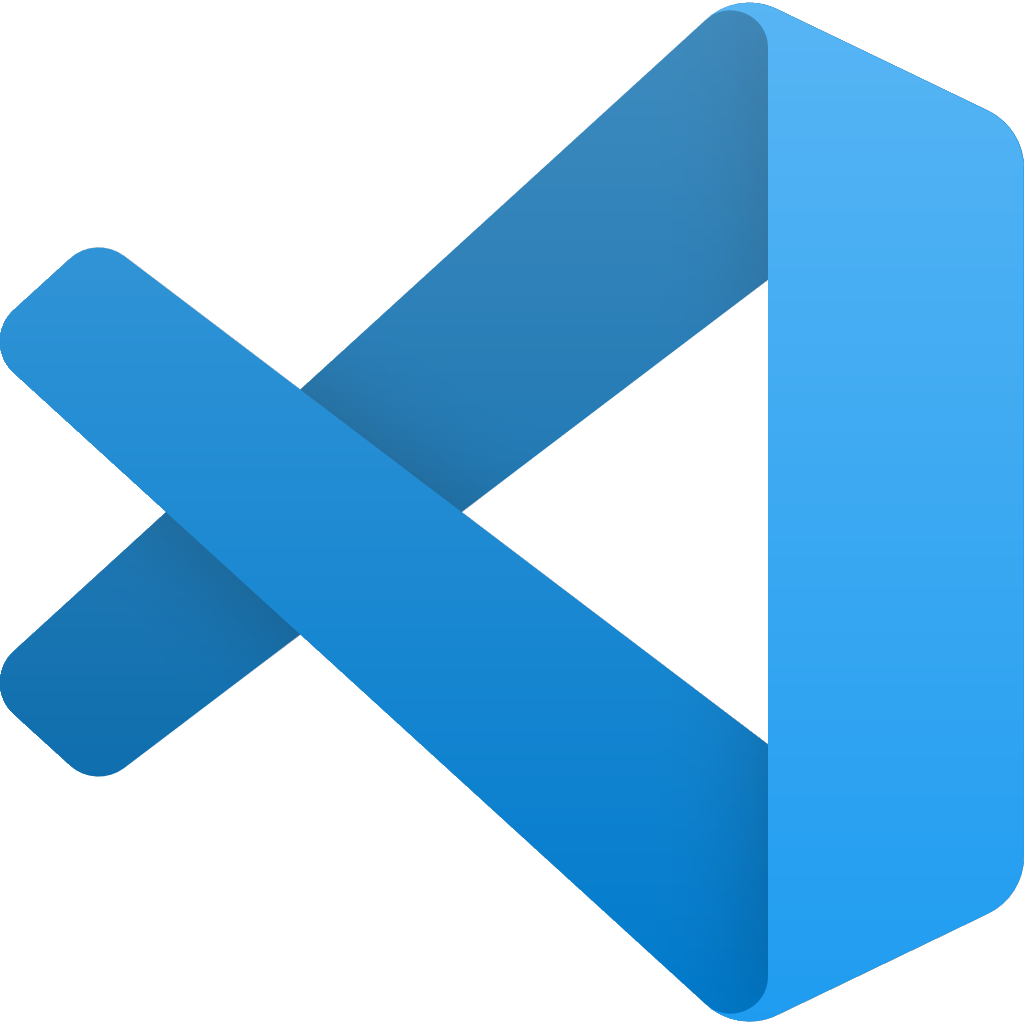


ภาพที่ 3.10 ภาพแสดงหน้าตัวอย่าง Server

**3.1.2 รายละเอียดของเครื่องมือที่ใช้**

3.1.2.1 Microsoft Visual Studio Code

เป็นโปรแกรมแก้ไข source-code ฟรีจาก Microsoft สามารถใช้แก้ไขโค้ดและเขียนโปรแกรมต่างๆได้ สามารถดาวโหลดส่วนประกอบเพิ่มเติมได้เช่นภาษาโปรแกรมอื่นๆ



ภาพที่ 3.11 Visual Studio Code

3.1.2.2 Switch Sound File Converter

เป็นโปรแกรมของ NCH Software ที่มีไว้แปลงไฟล์เสียงจากประเภทหนึ่งไปเป็นอีกประเภทหนึ่ง



ภาพที่ 3.12 Switch Sound File Converter

3.1.2.3 Google API: Speech to text

` เป็น API ของ Google ที่มีความสามารถในการแปลไฟล์เสียงเป็นข้อความได้

## 3.2 การจัดหา Dataset สำหรับทำการทดสอบ

**3.2.1 รายละเอียดของ Dataset**

เป็นไฟล์เสียงของนิทานเรื่องลูกหมูสามตัวมีความยาวประมาณ 4 หน้ากระดาษ A4

**3.2.2 การแปลงไฟล์**

ไฟล์เสียงที่ได้มาจะเป็น ไฟล์ MP3/M4A ซึ่งตัว API ที่ใช้ไม่รองรับจะต้องใช้เป็นไฟล์ประเภท WAV หรือ RAW เท่านั้น จึงต้องอาศัยโปรแกรมภายนอกในการแปลงประเภทไฟล์

## 3.3 การสร้างเกณฑ์สำหรับการตรวจสอบคุณภาพของหนังสือเสียง

**3.3.1 ความถูกต้อง**

3.3.1.1 อธิบายวิธีการออกแบบและใช้

มีการตรวจสอบว่าเสียงพูดที่ได้เมื่อมีการแปลงข้อมูลออกมาเป็น text แล้วนั้นตรงกับ Script ที่ใส่เข้าไป ไม่ผิดเพี้ยนหรือได้ความหมายที่ต่างออกไป

**3.3.2 ความเร็ว**

3.3.1.1 อธิบายวิธีการออกแบบและใช้

มีการตรวจสอบว่าเสียงพูดที่ได้ไม่เร็วเกินไปจนจับความหมายไม่ได้ หรือช้าเกินไปจนทำให้ฟังดูน่าเบื่อและไม่เหมาะสมต่อพัฒนาการทางด้านการเรียนรู้ของกลุ่มเป้าหมายซึ่งสอดคล้องต่อทฤษฎีอ้างอิง

**3.3.3 การวรรคตอน**

3.3.1.1 อธิบายวิธีการออกแบบและใช้

มีการตรวจสอบว่า วรรคตอนแต่ละแบบมีความยาวที่เหมาะสมต่อพัฒนาการทางด้านการเรียนรู้ของกลุ่มเป้าหมายซึ่งสอดคล้องต่อทฤษฎีอ้างอิง หรือการวรรคตอนที่ยาวนานเกินไปอาจจะทำให้ความหมายผิดเพี้ยนไปได้เช่นกัน

## 3.4 ภาพรวมการทำงานของระบบ

3.4.1 รับไฟล์เสียงของหนังสือเสียงและบทของไฟล์เสียง

3.4.2 นำไฟล์เสียงไปแปลงเป็นข้อความ

3.4.3 นำข้อความที่ได้ไปเปรียบเทียบกับบทของไฟล์เสียงและบันทึกผล

3.4.4 นำไฟล์เสียงไปตรวจสอบความเร็ว

3.4.5 นำผลลัพธ์ที่ได้ไปเปรียบเทียบกับความเร็วตามเกณฑ์ที่เราต้องการและบันทึกผล

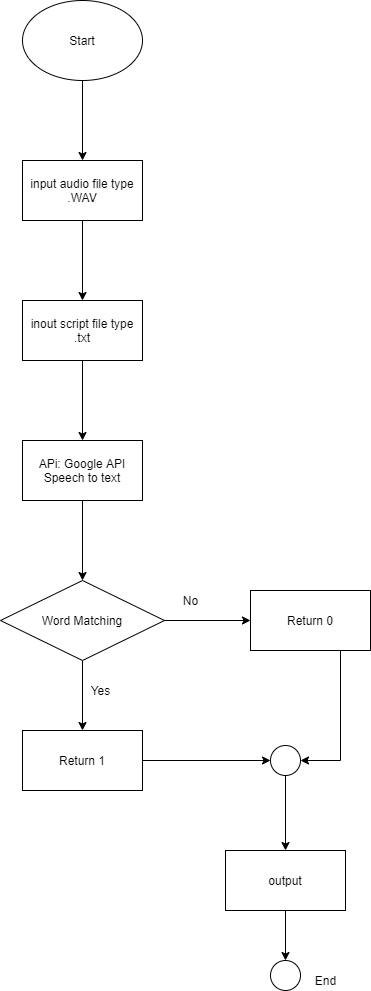
3.4.6 นำไฟล์เสียงที่ได้ไปตรวจสอบการเว้นวรรค

3.4.7 นำผลลัพธ์ที่ได้ไปเปรียบเทียบระยะเว้นวรรคตามเกณฑ์ที่เราต้องการและบันทึกผล

3.4.8 แสดงผลลัพธ์ที่ได้ทั้งหมด

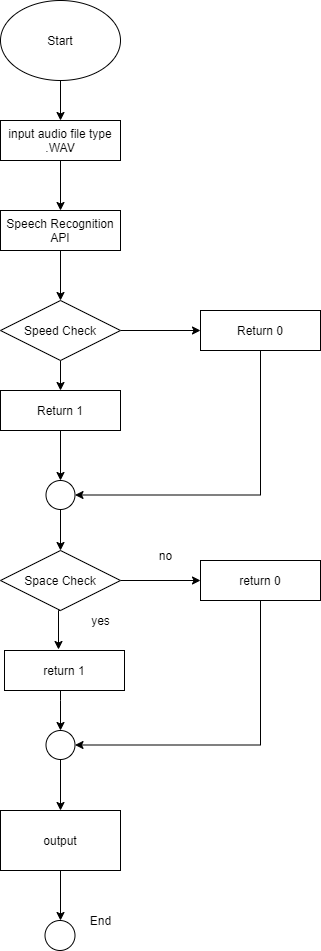
## 3.5 การออกแบบการทำงานและโครงสร้างของสถาปัตยกรรม

3.5.1 Word matching module



ภาพที่ 3.13 แสดงการทำงานของ Word matching module

3.5.2 Speech recognition module



ภาพที่ 3.14 แสดงการทำงานของ Speech recognition module

# บทที่ 4

**การทดลอง**

## 4.1 ทดสอบเดโม่ Word Matching Module กับตัวอย่างของ Google

**4.1.1 วัตถุประสงค์**

ทดสอบ demo ของ module word matching ว่าใช้งานได้กับไฟล์ตัวอย่างภาษาอังกฤษของ Google หรือไม่

**4.1.2 วิธีการทดสอบ**

4.1.2.1 นำไฟล์เสียงตัวอย่างที่จะใช้ทดสอบ จากภาพที่ 4.1 จะใช้ไฟล์ multi.wav ซึ่งเป็นไฟล์ทดสอบจาก Google ที่พ่วงมากับ API และ Script มาใส่ไว้ใน database ซึ่งที่นี้จะใช้เป็นไฟล์ json ตามภาพที่ 4.2

รูปภาพประกอบด้วย เมตร, ผู้เล่น

คำอธิบายที่สร้างโดยอัตโนมัติ

ภาพที่ 4.1 demo files

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

ภาพที่ 4.2 data.json

4.1.2.2 ใช้ฟังก์ชัน wordMatcher\_demo และใส่ index ของไฟล์ใน database ตามภาพที่ 4.3 เพื่อให้โปรแกรมเริ่มทำงาน โปรแกรมจะเรียกใช้ฟังก์ชัน transcribe\_file ภาพที่ 4.5

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

ภาพที่ 4.3 main.py



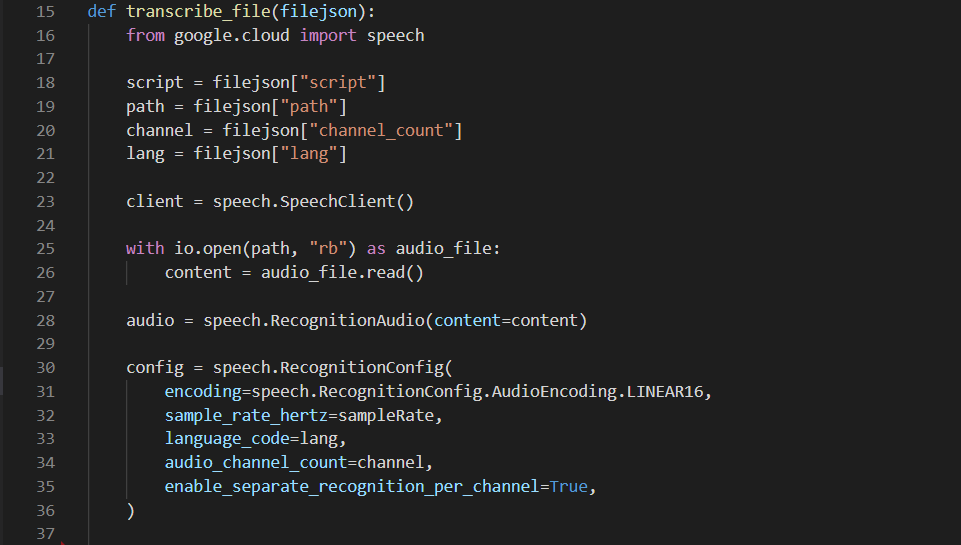
ภาพที่ 4.4 word\_matching.py

รูปภาพประกอบด้วย ข้อความ

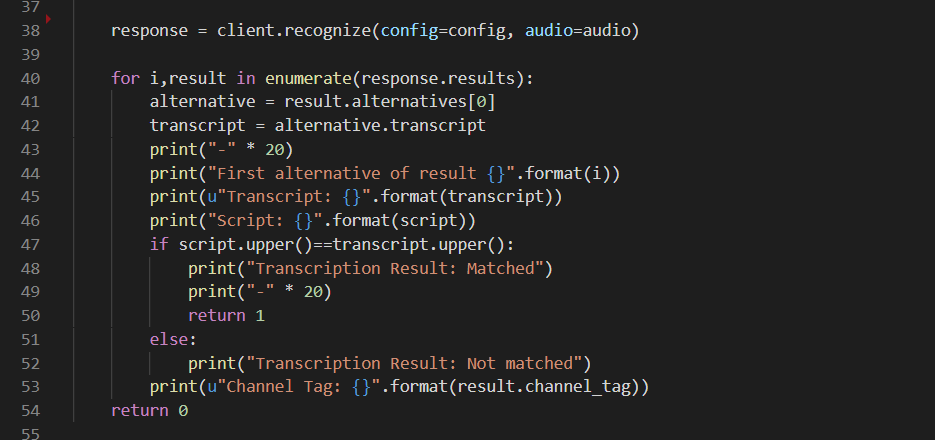
คำอธิบายที่สร้างโดยอัตโนมัติ

ภาพที่ 4.5 wordMatcher\_demo

4.1.2.3 transcribe\_file จะเรียก Speech to Text API ของ Google ออกมาเพื่อแปลงไฟล์เสียงให้เป็นข้อความตามภาพที่ 4.6 จากนั้นจะแสดงข้อความที่แปลได้แล้วนำมาเปรี่ยบเทียบกับบทที่ให้ไว้ว่าตรงกันหรือไม่ ถ้าตรงกันจะ print บน console ว่า Matched และ return ค่า 1 ออกมา แต่ถ้าไม่ตรงกันจะ print ว่า not Matched และ return ค่า 0 ตามภาพที่ 4.7



ภาพที่ 4.6 transcribe\_file 1



ภาพที่ 4.7 transcribe\_file 2

**4.1.3 ผลการทดสอบ**

ฟังก์ชัน return ค่า 1 และ print ว่า Matched บทการแปลที่ได้ความว่า “how are you doing estoy bien e tu” ซึ่งตรงกันกับบทที่ใส่ไว้ ตามภาพที่ 4.8

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

ภาพที่ 4.8 result 1

## 4.2 ทดสอบเดโม่ Word Matching Module กับตัวอย่างภาษาอังกฤษ

**4.2.1 วัตถุประสงค์**

ทดสอบ demo ของ module word matching ว่าใช้งานได้กับไฟล์ตัวอย่างภาษาอังกฤษที่อัดโดยผู้ทำโครงงานหรือไม่

**4.2.2 วิธีการทดสอบ**

4.2.2.1 นำไฟล์เสียงตัวอย่างที่จะใช้ทดสอบ จากภาพที่ 4.1 จะใช้ไฟล์ demoEng.wav ซึ่งเป็นไฟล์ทดสอบจาก Google ที่พ่วงมากับ API และ Script มาใส่ไว้ใน database ซึ่งที่นี้จะใช้เป็นไฟล์ json ตามภาพที่ 4.2

4.2.2.2 ใช้ฟังก์ชัน wordMatcher\_demo และใส่ index ของไฟล์ใน database ตามภาพที่ 4.3 เพื่อให้โปรแกรมเริ่มทำงาน โปรแกรมจะเรียกใช้ฟังก์ชัน transcribe\_file ภาพที่ 4.5

4.2.2.3 transcribe\_file จะเรียก Speech to Text API ของ Google ออกมาเพื่อแปลงไฟล์เสียงให้เป็นข้อความตามภาพที่ 4.6 จากนั้นจะแสดงข้อความที่แปลได้แล้วนำมาเปรี่ยบเทียบกับบทที่ให้ไว้ว่าตรงกันหรือไม่ ถ้าตรงกันจะ print บน console ว่า Matched และ return ค่า 1 ออกมา แต่ถ้าไม่ตรงกันจะ print ว่า not Matched และ return ค่า 0 ตามภาพที่ 4.7

**4.2.3 ผลการทดสอบ**

ฟังก์ชัน return ค่า 1 และ print ว่า Matched บทการแปลที่ได้ความว่า “how old is the Brooklyn Bridge” ซึ่งตรงกันกับบทที่ใส่ไว้ ตามภาพที่ 4.9

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

ภาพที่ 4.9 result 2

## 4.3 ทดสอบเดโม่ Word Matching Module กับตัวอย่างภาษาไทย

**4.3.1 วัตถุประสงค์**

ทดสอบ demo ของ module word matching ว่าใช้งานได้กับไฟล์ตัวอย่างภาษาไทยที่อัดโดยผู้ทำโครงงานหรือไม่

**4.3.2 วิธีการทดสอบ**

4.3.2.1 นำไฟล์เสียงตัวอย่างที่จะใช้ทดสอบ จากภาพที่ 4.1 จะใช้ไฟล์ demoTh.wav ซึ่งเป็นไฟล์ทดสอบจาก Google ที่พ่วงมากับ API และ Script มาใส่ไว้ใน database ซึ่งที่นี้จะใช้เป็นไฟล์ json ตามภาพที่ 4.2

4.3.2.2 ใช้ฟังก์ชัน wordMatcher\_demo และใส่ index ของไฟล์ใน database ตามภาพที่ 4.3 เพื่อให้โปรแกรมเริ่มทำงาน โปรแกรมจะเรียกใช้ฟังก์ชัน transcribe\_file ภาพที่ 4.5

4.3.2.3 transcribe\_file จะเรียก Speech to Text API ของ Google ออกมาเพื่อแปลงไฟล์เสียงให้เป็นข้อความตามภาพที่ 4.6 จากนั้นจะแสดงข้อความที่แปลได้แล้วนำมาเปรี่ยบเทียบกับบทที่ให้ไว้ว่าตรงกันหรือไม่ ถ้าตรงกันจะ print บน console ว่า Matched และ return ค่า 1 ออกมา แต่ถ้าไม่ตรงกันจะ print ว่า not Matched และ return ค่า 0 ตามภาพที่ 4.7

**4.3.3 ผลการทดสอบ**

ฟังก์ชัน return ค่า 1 และ print ว่า Matched บทการแปลที่ได้ความว่า “กาลครั้งหนึ่งนานมาแล้วมีลูกหมู 3 ตัวเป็นพี่น้องกันพวกมันออกเดินทางเพื่อจะหาที่สร้างบ้านคนละหลังเมื่อเดินทางมาถึงชายป่าแห่งพวกมันตัดสินใจจะสร้างบ้านบริเวณนี้ใกล้ๆกันและเดินทางไปตลาดเพื่อซื้อของสร้างบ้าน” ซึ่งตรงกันกับบทที่ใส่ไว้ ตามภาพที่ 4.10

รูปภาพประกอบด้วย ข้อความ

คำอธิบายที่สร้างโดยอัตโนมัติ

ภาพที่ 4.10 result 3

# บทที่ 5

**ข้อสรุปและข้อเสนอแนะ**

## 5.1 ข้อสรุป

5.1.1 ออกแบบเกณฑ์การวัดค่าความถูกต้องของหนังสือเสียงที่สอดคล้องกับผลงานวิจัยอ้างอิง

5.1.2 สามารถพัฒนา demo สำหรับในส่วน Word matching ให้สามารถใช้กับภาษาไทยได้และได้ผลลัพธ์ที่มีความถูกต้องในการตรวจสอบความถูกต้องของการสะกดคำ

5.1.3 สามารถจัดทำ dataset ได้บางส่วน

## 5.2 ข้อเสนอแนะ

ใช้ไฟล์เสียงประเภท Lossless audio เช่น M4Aหรือ WAV จะได้ไฟล์เสียงที่คุณภาพไม่ลดลงและจะลดเวลาในการดำเนินในการปรับปรุงคุณภาพไฟล์เสียง

## 5.3 ปัญหาและอุปสรรคที่พบ

ปัญหาด้าน dataset ที่จะต้องจัดทำเองให้ทำให้ใช้เวลาในการทำค่อนข้างนาน อีกทั้งยังมีความหลากหลายของคุณภาพของ dataset ที่ได้จึงต้องมีการปรับปรุงคุณภาพไฟล์เสียงและแปลงไฟล์

## 5.4 สิ่งที่จะทำต่อไป

5.4.1 ทำในส่วนของ Speech Recognition

5.4.1.1 ศึกษาการใช้งาน Speech Recognition API

5.4.1.2 พัฒนา demo สำหรับ Speech Recognition

5.4.1.3 ทดสอบและแก้ไขส่วนของ Speech Recognition

5.4.2 ทำในส่วนของ Word Matching

5.4.2.1 พัฒนาส่วนตรวจสอบความเร็วการอ่าน

5.4.2.2 พัฒนาส่วนของการตรวจสอบวรรคตอนการอ่าน

5.4.2.3 ทดสอบและแก้ไขส่วน Word matching

5.4.3 ทำในส่วนของ dataset

5.4.3.1 จัดหา dataset

5.4.3.2 ปรับปรุงคุณภาพไฟล์เสียงประเภท Lossy audio

5.4.4 รวมทั้ง 2 Module เข้าด้วยกัน

5.4.4.1 ออกแบบระบบวัดความถูกความต้องของหนังสือเสียง

5.4.4.2 พัฒนาระบบวัดความถูกต้องของหนังสือเสียง

5.4.4.3 ทดสอบและวัดประสิทธิภาพของตัวโปรแกรม

# บรรณานุกรม

Adam Geitgey,(2016),Machine Learning is Fun Part 5, “Language Translation with Deep Learning and the Magic of Sequences”,[Online],เข้าถึงได้จาก: <https://medium.com/@ageitgey/machine-learning-is-fun-part-5-language-translation-with-deep-learning-and-the-magic-of-sequences-2ace0acca0aa>

Adam Geitgey,(2016), Machine Learning is Fun Part 6, “How to do Speech Recognition with Deep Learning”,[Online],เข้าถึงได้จาก: <https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a>

Gamal Bohouta, Veton Z Këpuska (2017), “Comparing Speech Recognition Systems (Microsoft API, Google API And CMU Sphinx)”, Florida Institute of Technology,[Online],เข้าถึงได้จาก:<https://www.researchgate.net/publication/314938892_Comparing_Speech_Recognition_Systems_Microsoft_API_Google_API_And_CMU_Sphinx>

Nenny Anggraini, Angga Kuniawan, Luh Kesuma Wardhani, Nashrul Hakiem (2018),“Speech recognition application for the speech impaired using the android-based google cloud speech API”, Syarif Hidayatullah State Islamic University ,[Online],เข้าถึงได้จาก:Jakarta,<https://www.researchgate.net/publication/329865231_Speech_recognition_application_for_the_speech_impaired_using_the_android-based_google_cloud_speech_API>

[Carla A. Brooks](http://familyconnect.org/get-connected/about-familyconnect/contributing-experts/carla-a-brooks-ma-ccc-speech-language-pathologist/),Communication Skills for Children Who Are Blind or Visually Impaired,[Online],เข้าถึงได้จาก:<https://familyconnect.org/browse-by-age/infants-and-toddlers/growth-and-development-iandt/communication-skills-for-children-who-are-blind-or-visually-impaired/1235/>

[Lea M. McGee](https://www.readingrockets.org/articles/by-author/61886), [Judith Schickedanz](https://www.readingrockets.org/articles/by-author/61622),Repeated Interactive Read Alouds in Preschool and Kindergarten,[Online],เข้าถึงได้จาก:[https://www.readingrockets.org/article/repeated-interactive- read-alouds-preschool-and-kindergarten?fbclid=IwAR32\_bWDN09YzAYdqHCs0IMM9uLHBYmniPUMH5LsU-oUtB\_DxYZQSg87-4o](https://www.readingrockets.org/article/repeated-interactive-%20%20read-alouds-preschool-and-kindergarten?fbclid=IwAR32_bWDN09YzAYdqHCs0IMM9uLHBYmniPUMH5LsU-oUtB_DxYZQSg87-4o)

ไม่ปรากฏชื่อผู้เขียน,ชนิดของไฟล์เสียงบอกอะไรได้บ้าง,[Online],เข้าถึงได้จาก:https://www.mercular.com/review-article/audio-file-type?fbclid=IwAR3m2Z1j-6A15cgJfkulogoLypQBvzLASjiVtXrwSJc0Af3qUXGoobTDFOE

ไม่ปรากฏชื่อผู้เขียน,gRPC,[Online],เข้าถึงได้ จาก:https://en.wikipedia.org/wiki/GRPC?fbclid=IwAR0maXSHRnYmq4xPqkImwF1sYGN7ZnT8A4yPO9skdu587aX-mcflsrl9jGk

[saixiii](https://saixiii.com/author/saixiii/),(2017),RESTs,[Online],เข้าถึงได้จาก:<https://en.wikipedia.org/wiki/GRPC?fbclid=IwAR0maXSHRnYmq4xPqkImwF1sYGN7ZnT8A4yPO9skdu587aX-mcflsrl9jGk>

[saixiii](https://saixiii.com/author/saixiii/),(2017),API คืออะไร,[Online],เข้าถึงได้จาก:<https://saixiii.com/what-is-api/>

Google,Google API Speech to Text,[Online],เข้าถึงได้จาก:shorturl.at/diJQ1