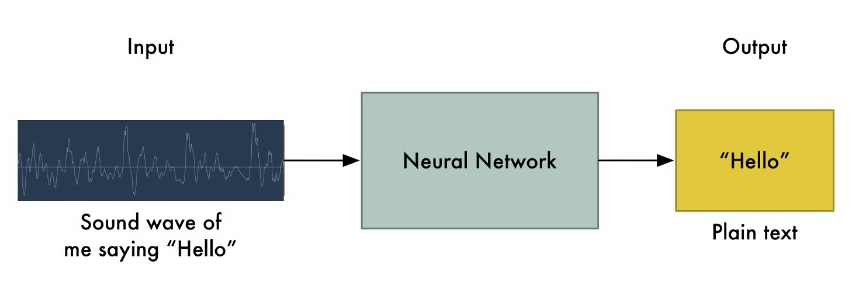
**บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง**

**2.1 Speech Recognition**

กระบวนการ Speech Recognition เป็นกระบวนการที่รับเอาเสียงพูดของมนุษย์ นำไปตีความและถอดความให้กลายเป็นตัวอักษร โดยวิธีการแบ่งเสียงพูดของมนุษย์ออกเป็นส่วนๆ และทำการวิเคราะห์เสียงแต่ละส่วน โดยการใช้ algorithms เพื่อหาคำที่เหมาะสมที่สุดกับภาษานั้นและทำการแปลงภาษาเหล่านั้นเป็นตัวอักษร โดยอาศัยกระบวนการดังนี้

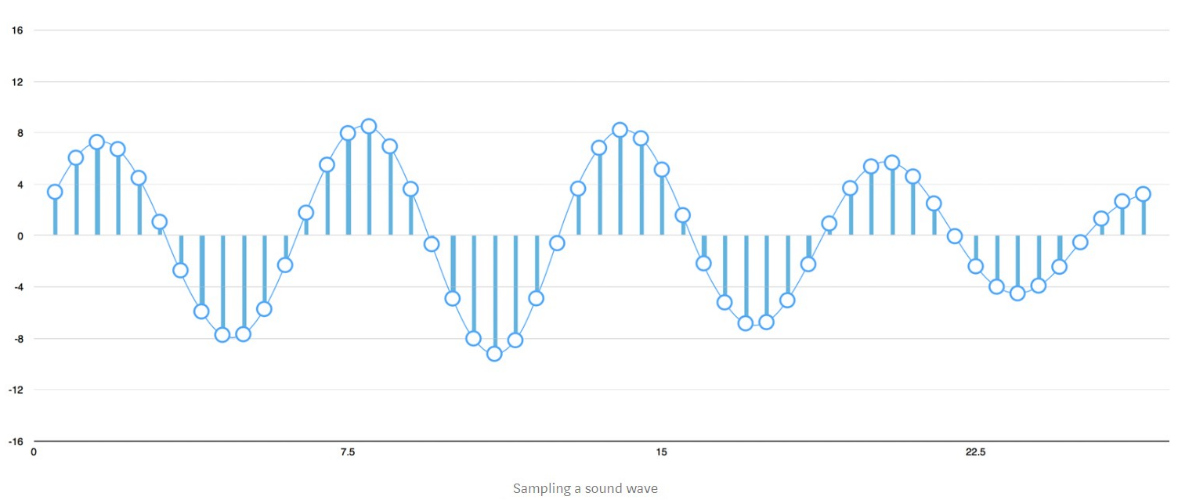
* ป้อนเสียงพูดที่บันทึก ลงใน neural network เพื่อแปลงเสียงพูดที่ถูกบันทึกไปเป็นข้อความ

simply feed sound recordings into a neural network and train to produce text

กระบวนการดังกล่าวมีปัญหาใหญ่อยู่นั่นก็คือ ความเร็วในการออกเสียงพูดของแต่ละคนไม่เท่ากัน บางคนพูดคำว่า “Hello” แบบเร็วๆ บางคนก็พูดคำว่า “Hello” แบบช้าๆ ซึ่งทำให้ขนาดของข้อมูลที่ถูกจัดเก็บมีขนาดไม่เท่ากัน ซึ่งตามหลักการแล้ว ไฟล์เสียงทั้งสองควรต้องจัดการให้เป็นข้อความเดียวกัน “Hello” ซึ่งการจัดการไฟล์เสียงทั้ง 2 ที่มีความยาวหลากหลายโดยอัตโนมัติให้เป็นข้อความที่มีความยาวคงที่นั้นทำได้ค่อนข้างยาก

เพื่อหลีกเลี่ยงปัญหาดังกล่าวจำเป็นต้องใช้เทคนิคพิเศษ และ processing เพิ่มเติมนอกเหนือจาก neural network กระบวนการที่แก้ไขปัญหามีขั้นตอนดังนี้

2.1.1 **Turning Sounds into Bits** ทำการป้อนคลื่นเสียงลงไปในคอมพิวเตอร์เพื่อเปลี่ยนคลื่นเสียงให้เป็นตัวเลข โดยวิธีการ sampling ทำการอ่านค่าเป็นพันๆ ครั้งต่อวินาที และบันทึกตัวเลขที่แสดงถึงความสูงของคลื่นเสียง ณ เวลานั้น



Sampled Digital Signal

2**.1.2. Pre-processing our Sampled Sound Data** ทำการจัดการกับข้อมูลที่ sampling มาและทำการแบ่งข้อมูลออกเป็นช่วงสั้นๆ แต่เนื่องจากข้อมูลเสียงนั้น มีความซับซ้อน เนื่องจากเป็นเสียงพูดของมนุษย์ซึ่งจะมีทั้งเสียงสูง เสียงกลาง เสียงต่ำ จึงต้องใช้เทคนิคพิเศษในการจำแนกเสียงและทำการแมพผลลัพธ์ตามความถี่ของเสียง (ต่ำ - สูง) และนำโมเดลที่ได้เข้าสู่ neural network ทำให้ neural network สามารถค้นหารูปแบบของข้อมูลประเภทนี้ได้ง่ายกว่าคลื่นเสียงแบบดังเดิม

2.1.3. **Recognizing Characters from Short Sounds** นำเอาท์พุทธ์ที่ได้จาก neural network ไปเป็นอินพุทธ์ให้กับ recurrent neural network (มีหน่วยความจำที่สามารถทำนายอนาคตได้) เนื่องจากตัวอักษรแต่ละตัวที่ทำนายควรมีผลต่อโอกาสของตัวอักษรถัดไปที่จะเกิดขึ้น เช่นหากพูดว่า "HEL" ก็จะมีโอกาสมากที่คำถัดไปจะพูดว่า "LO" และมีโอกาสน้อยที่จะพูดสิ่งที่ไม่สามารถออกเสียงได้ถัดไป เช่น "XYZ" ดังนั้นการมีหน่วยความจำที่เก็บค่าก่อนหน้านี้จะช่วยให้ neural network ทำการคาดการณ์ที่แม่นยำขึ้น

**2.2 Google Speech to Text API**

## **2.2.1 Speech requests**

Speech to Text มี 3 วิธีหลักในการจดจำเสียงพูด ดังต่อไปนี้  
 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 การรู้จำแบบ Synchronous Speech-to-Text API เป็นวิธีที่ง่ายที่สุดในการดำเนินการจดจำข้อมูลเสียงพูด Speech-to-Text สามารถประมวลผลข้อมูลเสียงพูดได้ถึง 1 นาทีที่ส่งใน request แบบ Synchronous หลังจากประมวลผล Speech-to-Text และจดจำเสียงทั้งหมดแล้วระบบจะส่งกลับการตอบสนอง

request แบบ Synchronous ถูกบล็อกซึ่งหมายความว่า Speech-to-Text ต้องส่งคืนการตอบกลับก่อนที่จะประมวลผล request ถัดไป โดยทั่วไป Speech-to-Text จะประมวลผลเสียงได้เร็วกว่าแบบเรียลไทม์โดยจะประมวลผลเสียง 30 วินาทีใน 15 วินาทีโดยเฉลี่ย ในกรณีที่คุณภาพเสียงไม่ดี request การจดจำของคุณอาจใช้เวลานานกว่ามาก

Speech-to-Text มีทั้งวิธี REST และ gRPC สำหรับการเรียกใช้ Speech-to-Text API แบบ Synchronous และ Asynchronous บทความนี้สาธิต REST API เนื่องจากแสดงและอธิบายการใช้งานพื้นฐานของ API ได้ง่ายกว่า อย่างไรก็ตามการแต่งหน้าพื้นฐานของ request REST หรือ gRPC นั้นค่อนข้างคล้ายกัน request การรับรู้การสตรีมได้รับการสนับสนุนโดย gRPC เท่านั้น

request การรู้จำเสียงแบบSynchronous

request API คำพูดเป็นข้อความแบบSynchronousประกอบด้วยการกำหนดค่าการรู้จำเสียงและข้อมูลเสียง request ตัวอย่างแสดงอยู่ด้านล่าง:

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

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

encoding - (จำเป็น) ระบุรูปแบบการเข้ารหัสของเสียงที่ให้มา (ประเภท AudioEncoding) หากคุณมีทางเลือกในตัวแปลงสัญญาณให้เลือกเข้ารหัสแบบไม่สูญเสียเช่น FLAC หรือ LINEAR16 เพื่อประสิทธิภาพที่ดีที่สุด (สำหรับข้อมูลเพิ่มเติมโปรดดูการเข้ารหัสเสียง) ฟิลด์การเข้ารหัสเป็นทางเลือกสำหรับไฟล์ FLAC และ WAV ที่การเข้ารหัสรวมอยู่ในส่วน file header

• sampleRateHertz - (จำเป็น) ระบุ samplerate (ในเฮิรตซ์) ของเสียงที่ให้มา (สำหรับข้อมูลเพิ่มเติมเกี่ยวกับ samplerate โปรดดู samplerate ด้านล่าง) ฟิลด์ • sampleRateHertz เป็นทางเลือกสำหรับไฟล์ FLAC และ WAV ซึ่ง samplerate จะรวมอยู่ในส่วน file header

• languageCode - (จำเป็น) มีภาษา + ภูมิภาค / สถานที่ที่จะใช้สำหรับการรู้จำเสียงของเสียงที่ให้มา รหัสภาษาต้องเป็นตัวระบุ BCP-47 โปรดทราบว่าโดยทั่วไปรหัสภาษาประกอบด้วยแท็กภาษาหลักและแท็กย่อยของภูมิภาครองเพื่อระบุภาษาถิ่น (ตัวอย่างเช่น 'en' สำหรับภาษาอังกฤษและ 'US' สำหรับสหรัฐอเมริกาในตัวอย่างข้างต้น) (สำหรับรายการภาษาที่รองรับโปรดดูที่รองรับ ภาษา)

• maxAlternatives - (ไม่บังคับค่าเริ่มต้นคือ 1) ระบุจำนวนการถอดเสียงทางเลือกที่จะให้ในการตอบกลับ ตามค่าเริ่มต้น Speech-to-Text API จะมีการถอดเสียงหลักหนึ่งรายการ หากคุณต้องการประเมินทางเลือกอื่นให้ตั้งค่า maxAlternatives เป็นค่าที่สูงขึ้น โปรดทราบว่า Speech-to-Text จะส่งคืนทางเลือกอื่นก็ต่อเมื่อตัวจดจำกำหนดทางเลือกที่มีคุณภาพเพียงพอ โดยทั่วไปทางเลือกอื่นจะเหมาะสมกว่าสำหรับ request แบบเรียลไทม์ที่ต้องการความคิดเห็นของผู้ใช้ (เช่นคำสั่งเสียง) ดังนั้นจึงเหมาะสำหรับ request การจดจำการสตรีมมากกว่า

• profanityFilter - (ไม่บังคับ) ระบุว่าจะกรองคำหรือวลีที่หยาบคายออกไป คำที่กรองออกจะมีอักษรตัวแรกและเครื่องหมายดอกจันสำหรับอักขระที่เหลือ (เช่น f \*\*\*) ตัวกรองคำหยาบคายทำงานในคำเดียวโดยไม่ตรวจพบคำพูดที่ไม่เหมาะสมหรือไม่เหมาะสมที่เป็นวลีหรือคำรวมกัน

• speechContext - (ไม่บังคับ) มีข้อมูลบริบทเพิ่มเติมสำหรับการประมวลผลเสียงนี้ บริบทประกอบด้วยฟิลด์ย่อยต่อไปนี้:

• comtext- มีรายการคำและวลีที่ให้คำแนะนำเกี่ยวกับงานการรู้จำเสียง สำหรับข้อมูลเพิ่มเติมโปรดดูข้อมูลเกี่ยวกับบริบทคำพูด

เสียงจะถูกส่งไปยัง Speech-to-Text ผ่านพารามิเตอร์เสียงประเภท RecognitionAudio ฟิลด์เสียงประกอบด้วยฟิลด์ย่อยต่อไปนี้:

• content - ประกอบด้วยเสียงในการประเมินซึ่งฝังอยู่ใน request ดูการฝังเนื้อหาเสียงด้านล่างสำหรับข้อมูลเพิ่มเติม เสียงที่ส่งผ่านโดยตรงภายในช่องนี้ จำกัด ระยะเวลาไว้ที่ 1 นาที

• uri มี URI ที่ชี้ไปที่เนื้อหาเสียง ต้องไม่บีบอัดไฟล์ (เช่น gzip) ปัจจุบันช่องนี้ต้องมี URI ของ Google Cloud Storage (ในรูปแบบ gs: // bucket-name / path\_to\_audio\_file) ดูการอ้างอิงการส่งผ่านเสียงโดย URI ด้านล่าง)

ข้อมูลเพิ่มเติมเกี่ยวกับ request และพารามิเตอร์การตอบกลับเหล่านี้ปรากฏด้านล่าง

**Sample rates**

คุณระบุ samplerate ของเสียงของคุณในฟิลด์ sampleRateHertz ของ configuration คำร้องขอและต้องตรงกับ samplerate ของเนื้อหาเสียงหรือสตรีมที่เกี่ยวข้อง samplerate ระหว่าง 8000 Hz ถึง 48000 Hz ได้รับการสนับสนุนภายใน Speech-to-Text samplerate สำหรับไฟล์ FLAC หรือ WAV สามารถกำหนดได้จากส่วนหัวของไฟล์แทนที่จะมาจากฟิลด์ sampleRateHertz

หากคุณมีทางเลือกในการเข้ารหัสวัสดุต้นทางให้จับเสียงโดยใช้ samplerate 16000 Hz ค่าที่ต่ำกว่านี้อาจทำให้ความแม่นยำในการรู้จำเสียงลดลงและระดับที่สูงขึ้นจะไม่มีผลต่อคุณภาพการรู้จำเสียง

อย่างไรก็ตามหากข้อมูลเสียงของคุณได้รับการบันทึกใน samplerate ที่มีอยู่แล้วที่นอกเหนือจาก 16000 Hz อย่าสุ่มตัวอย่างเสียงของคุณเป็น 16000 Hz ตัวอย่างเช่นเสียงโทรศัพท์ในระบบเดิมส่วนใหญ่ใช้ samplerate 8000 Hz ซึ่งอาจให้ผลลัพธ์ที่แม่นยำน้อยกว่า หากคุณต้องใช้เสียงดังกล่าวให้ส่งเสียงไปยัง Speech API ตาม samplerate ดั้งเดิม

**Languages**

เครื่องมือจดจำเสียงพูดเป็นข้อความรองรับภาษาและภาษาถิ่นที่หลากหลาย คุณระบุภาษา (และภาษาประจำชาติหรือภูมิภาค) ของเสียงของคุณภายในฟิลด์ languageCode ของการกำหนดค่า request โดยใช้ตัวระบุ BCP-47

รายการภาษาที่รองรับทั้งหมดสำหรับแต่ละคุณสมบัติมีอยู่ในหน้าการสนับสนุนภาษา

**Time offsets (timestamps)**

Speech-to-Text สามารถรวมค่าการชดเชยเวลา (การประทับเวลา) สำหรับจุดเริ่มต้นและจุดสิ้นสุดของคำพูดแต่ละคำที่รับรู้ในเสียงที่ให้มา ค่าชดเชยเวลาแสดงถึงระยะเวลาที่ผ่านไปจากจุดเริ่มต้นของเสียงโดยเพิ่มขึ้นทีละ 100 มิลลิวินาที

การชดเชยเวลามีประโยชน์อย่างยิ่งสำหรับการวิเคราะห์ไฟล์เสียงที่ยาวขึ้นซึ่งคุณอาจต้องค้นหาคำใดคำหนึ่งในข้อความที่รู้จักและค้นหาคำนั้น (ค้นหา) ในเสียงต้นฉบับ การชดเชยเวลาได้รับการสนับสนุนสำหรับวิธีการจดจำทั้งหมดของเรา: รับรู้สตรีมจดจำและจดจำระยะยาว

ค่าชดเชยเวลาจะรวมไว้สำหรับทางเลือกแรกที่ระบุในการตอบสนองการรับรู้เท่านั้น

หากต้องการรวมการชดเชยเวลาในผลลัพธ์ของ request ของคุณให้ตั้งค่าพารามิเตอร์ enableWordTimeOffsets เป็น true ใน configuration request ของคุณ สำหรับตัวอย่างการใช้ REST API หรือไลบรารีไคลเอ็นต์โปรดดูการใช้ Time Offsets (Timestamps) ตัวอย่างเช่นคุณสามารถรวมพารามิเตอร์ enableWordTimeOffsets ใน configuration request ดังที่แสดงไว้ที่นี่:

{  
"config": {  
  "languageCode": "en-US",  
  "enableWordTimeOffsets": true  
  },  
"audio":{  
  "uri":"gs://gcs-test-data/gettysburg.flac"  
  }  
}

ผลลัพธ์ที่ส่งคืนโดย Speech-to-Text API จะมีค่าชดเชยเวลาสำหรับแต่ละคำที่รู้จักดังที่แสดงต่อไปนี้

  "name": "6212202767953098955",  
  "metadata": {  
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeMetadata",  
    "progressPercent": 100,  
    "startTime": "2017-07-24T10:21:22.013650Z",  
    "lastUpdateTime": "2017-07-24T10:21:45.278630Z"  
  },  
  "done": true,  
  "response": {  
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeResponse",  
    "results": [  
      {  
        "alternatives": [  
          {  
            "transcript": "Four score and twenty...(etc)...",  
            "confidence": 0.97186122,  
            "words": [  
              {  
                "startTime": "1.300s",  
                "endTime": "1.400s",  
                "word": "Four"  
              },  
              {  
                "startTime": "1.400s",  
                "endTime": "1.600s",  
                "word": "score"  
              },  
              {  
                "startTime": "1.600s",  
                "endTime": "1.600s",  
                "word": "and"  
              },  
              {  
                "startTime": "1.600s",  
                "endTime": "1.900s",  
                "word": "twenty"  
              },  
              ...  
            ]  
          }  
        ]  
      },  
      {  
        "alternatives": [  
          {  
            "transcript": "for score and plenty...(etc)...",  
            "confidence": 0.9041967,  
          }  
        ]  
      }  
    ]  
  }  
}

**การเลือกรูปแบบโมเดล**

Speech-to-Text สามารถใช้โมเดลการเรียนรู้ของเครื่องหลายรูปแบบเพื่อถอดเสียงไฟล์เสียงของคุณ Google ได้ฝึกอบรมรูปแบบการรู้จำเสียงเหล่านี้สำหรับประเภทเสียงและแหล่งที่มาเฉพาะ

เมื่อคุณส่ง request การถอดเสียงเป็นคำพูดเป็นข้อความคุณสามารถปรับปรุงผลลัพธ์ที่ได้รับโดยระบุแหล่งที่มาของเสียงต้นฉบับ สิ่งนี้ช่วยให้ Speech-to-Text API ประมวลผลไฟล์เสียงของคุณโดยใช้โมเดล machine learning ที่ได้รับการฝึกฝนให้จดจำเสียงพูดจากแหล่งข้อมูลประเภทนั้น ๆ

ในการระบุโมเดลสำหรับการรู้จำเสียงให้รวมฟิลด์โมเดลในอ็อบเจ็กต์ RecognitionConfig สำหรับ request ของคุณโดยระบุโมเดลที่คุณต้องการใช้

Speech-to-Text สามารถใช้โมเดล machine learning ประเภทต่อไปนี้สำหรับการถอดเสียงไฟล์เสียงของคุณ

หมายเหตุ: ดูหน้าภาษาที่รองรับเพื่อดูว่าภาษาของคุณมีรุ่นใดบ้าง

|  |  |  |
| --- | --- | --- |
| **Type** | **Enum constant** | **Description** |
| Video | video | Use this model for transcribing audio in video clips or that includes multiple speakers. For best results, provide audio recorded at 16,000Hz or greater sampling rate.  **Note:** This is a premium model that costs more than the standard rate. |
| Phone call | phone\_call | Use this model for transcribing audio from a phone call. Typically, phone audio is recorded at 8,000Hz sampling rate.  **Note:** The enhanced phone model is a premium model that costs more than the standard rate. |
| ASR: Command and search | command\_and\_search | Use this model for transcribing shorter audio clips. Some examples include voice commands or voice search. |
| ASR: Default | default | Use this model if your audio does not fit one of the previously described models. For example, you can use this for long-form audio recordings that feature a single speaker only. Ideally, the audio is high-fidelity, recorded at 16,000Hz or greater sampling rate. |

### **Embedding audio content**

การฝังเนื้อหาเสียง

เสียงที่ฝังไว้จะรวมอยู่ใน request การรู้จำเสียงเมื่อส่งผ่านพารามิเตอร์เนื้อหาภายในช่องเสียงของ request สำหรับเสียงที่ฝังไว้ให้เป็นเนื้อหาภายในคำร้องขอ gRPC เสียงนั้นจะต้องเข้ากันได้สำหรับการทำให้เป็นอนุกรม Proto3 และจัดให้เป็นข้อมูลไบนารี สำหรับเสียงที่ฝังไว้ที่จัดเตรียมไว้เป็นเนื้อหาภายใน request REST เสียงนั้นจะต้องเข้ากันได้กับการทำให้เป็นอนุกรม JSON และอันดับแรกต้องเข้ารหัส Base64 ดู Base64 การเข้ารหัสเสียงของคุณสำหรับข้อมูลเพิ่มเติม)

เมื่อสร้าง request โดยใช้ไลบรารีไคลเอ็นต์ Google Cloud โดยทั่วไปคุณจะเขียนข้อมูลไบนารี (หรือฐาน 64 ที่เข้ารหัส) ลงในช่องเนื้อหาโดยตรง

**ส่งผ่านเสียงที่อ้างอิงโดย URI**

โดยทั่วไปคุณจะส่งพารามิเตอร์ uri ภายในช่องเสียงของ request Speech โดยชี้ไปที่ไฟล์เสียง (ในรูปแบบไบนารีไม่ใช่ base64) ที่อยู่บน Google Cloud Storage ในรูปแบบต่อไปนี้

gs://*bucket-name*/*path\_to\_audio\_file*

ตัวอย่างเช่นส่วนต่อไปนี้ของ request Speech อ้างอิงไฟล์เสียงตัวอย่างที่ใช้ภายใน Quickstart:...

    "audio": {  
        "uri":"gs://cloud-samples-tests/speech/brooklyn.flac"  
    }  
...

คุณต้องมีสิทธิ์การเข้าถึงที่เหมาะสมเพื่ออ่านไฟล์ Google Cloud Storage เช่นอย่างใดอย่างหนึ่งต่อไปนี้:

•อ่านได้แบบสาธารณะ (เช่นไฟล์เสียงตัวอย่างของเรา)

•สามารถอ่านได้จากบัญชีบริการของคุณหากใช้การอนุญาตบัญชีบริการ

•สามารถอ่านได้โดยบัญชีผู้ใช้หากใช้ OAuth แบบ 3 ทางสำหรับการอนุญาตบัญชีผู้ใช้

ดูข้อมูลเพิ่มเติมเกี่ยวกับการจัดการการเข้าถึง Google Cloud Storage ได้ที่การสร้างและจัดการรายการควบคุมการเข้าถึงในเอกสาร Google Cloud Storage การตอบสนองของ Speech-to-Text API ตามที่ระบุไว้ก่อนหน้านี้การตอบสนอง API แบบ Speech-to-Text แบบSynchronousอาจใช้เวลาสักครู่เพื่อส่งคืนผลลัพธ์ตามสัดส่วนกับความยาวของเสียงที่ให้ เมื่อประมวลผลแล้ว API จะตอบกลับตามที่แสดงด้านล่าง:

{  
  "results": [  
    {  
      "alternatives": [  
        {  
          "confidence": 0.98267895,  
          "transcript": "how old is the Brooklyn Bridge"  
        }  
      ]  
    }  
  ]  
}

ฟิลด์เหล่านี้มีคำอธิบายด้านล่าง:

•ผลลัพธ์ประกอบด้วยรายการผลลัพธ์ (ประเภท SpeechRecognitionResult) โดยแต่ละผลลัพธ์จะสอดคล้องกับส่วนของเสียง (ส่วนของเสียงจะถูกคั่นด้วยการหยุดชั่วคราว) ผลลัพธ์แต่ละรายการจะประกอบด้วยฟิลด์ต่อไปนี้อย่างน้อยหนึ่งฟิลด์:

•ทางเลือกประกอบด้วยรายการการถอดเสียงที่เป็นไปได้ประเภท SpeechRecognitionAlternatives ทางเลือกมากกว่าหนึ่งทางจะปรากฏขึ้นหรือไม่นั้นขึ้นอยู่กับว่าคุณร้องขอมากกว่าหนึ่งทางเลือก (โดยการตั้งค่า maxAlternatives เป็นค่าที่มากกว่า 1) และดูว่าทางเลือกที่สร้างจาก Speech-to-Text มีคุณภาพสูงเพียงพอหรือไม่ แต่ละทางเลือกจะประกอบด้วยฟิลด์ต่อไปนี้:

•การถอดเสียงมีข้อความที่ถอดเสียง ดูการจัดการการถอดเสียงด้านล่าง

•ค่าความเชื่อมั่นมีค่าระหว่าง 0 ถึง 1 ที่บ่งบอกว่าการถอดเสียงพูดเป็นข้อความนั้นมีความมั่นใจเพียงใด ดูการตีความค่าความเชื่อมั่นด้านล่าง

หากไม่สามารถจดจำเสียงพูดจากเสียงที่ให้มาได้รายการผลลัพธ์ที่ส่งคืนจะไม่มีรายการใด ๆ คำพูดที่ไม่รู้จักมักเป็นผลมาจากเสียงที่มีคุณภาพต่ำมากหรือจากค่ารหัสภาษาการเข้ารหัสหรืออัตราตัวอย่างที่ไม่ตรงกับเสียงที่ให้มา

ส่วนประกอบของการตอบกลับนี้อธิบายไว้ในส่วนต่อไปนี้

การตอบสนอง API แบบ Speech-to-Text แบบซิงโครนัสแต่ละรายการจะแสดงรายการผลลัพธ์แทนที่จะเป็นผลลัพธ์เดียวที่มีเสียงที่รู้จักทั้งหมด รายการเสียงที่รู้จัก (ภายในองค์ประกอบการถอดเสียง) จะปรากฏตามลำดับที่ต่อเนื่องกัน

การเลือกทางเลือกอื่น

ผลลัพธ์แต่ละรายการภายในการตอบสนองการรับรู้แบบซิงโครนัสที่ประสบความสำเร็จสามารถมีทางเลือกอย่างน้อยหนึ่งทางเลือก (หากค่า maxAlternatives สำหรับคำขอมากกว่า 1) หาก Speech-to-Text ระบุว่าทางเลือกหนึ่งมีค่าความเชื่อมั่นที่เพียงพอทางเลือกนั้นจะรวมอยู่ในคำตอบ ทางเลือกแรกในการตอบสนองคือทางเลือกที่ดีที่สุด (เป็นไปได้มากที่สุด) เสมอ

การตั้งค่า maxAlternatives เป็นค่าที่สูงกว่า 1 ไม่ได้หมายความหรือรับประกันว่าจะส่งคืนทางเลือกหลายทาง โดยทั่วไปมีทางเลือกมากกว่าหนึ่งทางเลือกที่เหมาะสมกว่าสำหรับการให้ตัวเลือกแบบเรียลไทม์แก่ผู้ใช้เพื่อรับผลลัพธ์ผ่านคำขอการจดจำสตรีมมิง

**Selecting alternatives**  
ผลลัพธ์แต่ละรายการภายในการตอบสนองการจดจำแบบซิงโครนัสที่ประสบความสำเร็จสามารถมีทางเลือกอย่างน้อยหนึ่งรายการ (หากค่า maxAlternatives สำหรับคำขอมากกว่า 1) หาก Speech-to-Text ระบุว่าทางเลือกมีค่าความเชื่อมั่นที่เพียงพอทางเลือกนั้นจะรวมอยู่ในการตอบกลับ ทางเลือกแรกในการตอบสนองคือทางเลือกที่ดีที่สุด (เป็นไปได้มากที่สุด) เสมอ การตั้งค่า maxAlternatives เป็นค่าที่สูงกว่า 1 ไม่ได้หมายความหรือรับประกันว่าจะส่งคืนทางเลือกหลายทาง โดยทั่วไปทางเลือกมากกว่าหนึ่งทางเลือกที่เหมาะสมกว่าสำหรับการให้ตัวเลือกแบบเรียลไทม์แก่ผู้ใช้ที่ได้รับผลลัพธ์ผ่านคำขอการจดจำสตรีมมิง

**Handling transcriptions**

ทางเลือกแต่ละทางที่ให้มาในคำตอบจะมีการถอดเสียงที่มีข้อความที่รู้จัก เมื่อให้ทางเลือกตามลำดับคุณควรเชื่อมการถอดเสียงเหล่านี้เข้าด้วยกัน

โค้ด Python ต่อไปนี้จะวนซ้ำเหนือรายการผลลัพธ์และเชื่อมต่อการถอดเสียงเข้าด้วยกัน โปรดทราบว่าเราใช้ทางเลือกแรก (zeroth)ในทุกกรณี

response = service\_request.execute()  
recognized\_text = 'Transcribed Text: \n'  
for i in range(len(response['results'])):  
    recognized\_text += response['results'][i]['alternatives'][0]['transcript']

### **Confidence values**

ค่าความเชื่อมั่นคือค่าประมาณระหว่าง 0.0 ถึง 1.0 คำนวณโดยการรวมค่า "ความเป็นไปได้" ที่กำหนดให้กับแต่ละคำในเสียง ตัวเลขที่สูงกว่าแสดงถึงความเป็นไปได้ที่มากขึ้นโดยประมาณที่จำแต่ละคำได้อย่างถูกต้อง โดยทั่วไปฟิลด์นี้มีไว้สำหรับสมมติฐานด้านบนเท่านั้นและสำหรับผลลัพธ์ที่ is\_final = true เท่านั้น ตัวอย่างเช่นคุณอาจใช้ค่าความเชื่อมั่นเพื่อตัดสินใจว่าจะแสดงผลลัพธ์อื่นให้กับผู้ใช้หรือขอการยืนยันจากผู้ใช้ อย่างไรก็ตามโปรดทราบว่าแบบจำลองจะกำหนดผลลัพธ์ที่ "ดีที่สุด" ซึ่งอยู่ในอันดับสูงสุดโดยอาศัยสัญญาณมากกว่าคะแนนความเชื่อมั่นเพียงอย่างเดียว (เช่นบริบทของประโยค) ด้วยเหตุนี้จึงมีบางกรณีที่ผลลัพธ์อันดับต้น ๆ ไม่มีคะแนนความเชื่อมั่นสูงสุด หากคุณไม่ได้ร้องขอผลลัพธ์ทางเลือกหลายรายการผลลัพธ์ที่ "ดีที่สุด" เดียวที่ส่งคืนอาจมีค่าความเชื่อมั่นต่ำกว่าที่คาดการณ์ไว้ สิ่งนี้อาจเกิดขึ้นได้เช่นในกรณีที่มีการใช้คำที่หายาก คำที่ไม่ค่อยได้ใช้สามารถกำหนดค่า "ความเป็นไปได้" ที่ต่ำแม้ว่าจะจำได้ถูกต้องก็ตาม หากโมเดลกำหนดให้คำหายากเป็นตัวเลือกที่เป็นไปได้มากที่สุดตามบริบทผลลัพธ์นั้นจะถูกส่งกลับที่ด้านบนแม้ว่าค่าความเชื่อมั่นของผลลัพธ์จะต่ำกว่าตัวเลือกอื่นก็ตาม หมายเหตุ: รหัสของคุณไม่ควรคาดหวังความเชื่อมั่นเป็นฟิลด์บังคับเนื่องจากไม่รับประกันว่าจะถูกต้องหรือแม้กระทั่งตั้งค่าในผลลัพธ์ใด ๆ คำขอและการตอบกลับแบบอะซิงโครนัส คำขอ API แบบอะซิงโครนัส Speech-to-Text ไปยังเมธอด LongRunningRecognize นั้นเหมือนกันในรูปแบบของคำขอ API แบบ Speech-to-Text แบบซิงโครนัส อย่างไรก็ตามแทนที่จะส่งคืนการตอบกลับคำขอแบบอะซิงโครนัสจะเริ่มต้นการดำเนินการแบบ Long Running Operation (ประเภท Operation) และส่งคืนการดำเนินการนี้ไปยัง callee ทันที การตอบสนองการทำงานทั่วไปแสดงไว้ด้านล่าง:{

  "name": "*operation\_name*",  
  "metadata": {  
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeMetadata"  
    "progressPercent": 34,  
    "startTime": "2016-08-30T23:26:29.579144Z",  
    "lastUpdateTime": "2016-08-30T23:26:29.826903Z"  
  }  
}

โปรดทราบว่ายังไม่มีผลลัพธ์ Speech-to-Text จะประมวลผลเสียงที่ให้มาต่อไปและใช้การดำเนินการนี้เพื่อจัดเก็บผลลัพธ์สุดท้ายซึ่งจะปรากฏในฟิลด์การตอบสนองของการดำเนินการ (ประเภท LongRunningRecognizeResponse) เมื่อเสร็จสิ้นคำขอ

การตอบกลับทั้งหมดหลังจากเสร็จสิ้นคำขอจะปรากฏด้านล่าง:

{  
  "name": "1268386125834704889",  
  "metadata": {  
    "lastUpdateTime": "2016-08-31T00:16:32.169Z",  
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongrunningRecognizeMetadata",  
    "startTime": "2016-08-31T00:16:29.539820Z",  
    "progressPercent": 100  
  }  
  "response": {  
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeResponse",  
    "results": [{  
      "alternatives": [{  
        "confidence": 0.98267895,  
        "transcript": "how old is the Brooklyn Bridge"  
      }]}]  
  },  
  "done": True,  
}

โปรดทราบว่าเสร็จแล้วได้รับการตั้งค่าเป็น True และการตอบสนองของการดำเนินการประกอบด้วยชุดของผลลัพธ์ประเภท SpeechRecognitionResult ซึ่งเป็นประเภทเดียวกันที่ส่งคืนโดยคำขอการรู้จำ Speech-to-Text แบบซิงโครนัส

ตามค่าเริ่มต้นการตอบสนอง REST แบบอะซิงโครนัสจะถูกตั้งค่าเป็น False ซึ่งเป็นค่าเริ่มต้น อย่างไรก็ตามเนื่องจาก JSON ไม่ต้องการให้มีค่าเริ่มต้นภายในฟิลด์เมื่อทดสอบว่าการดำเนินการเสร็จสมบูรณ์หรือไม่คุณควรทดสอบทั้งสองฟิลด์ว่ามีอยู่หรือไม่และตั้งค่าเป็น True

การสตรีมคำขอการรู้จำ Speech-to-Text API

การโทรการรู้จำ Speech-to-Text แบบสตรีมมิ่งได้รับการออกแบบมาสำหรับการบันทึกและจดจำเสียงแบบเรียลไทม์ภายในสตรีมแบบสองทิศทาง แอปพลิเคชันของคุณสามารถส่งเสียงในสตรีมคำขอและรับผลการจดจำระหว่างกาลและขั้นสุดท้ายในสตรีมการตอบกลับแบบเรียลไทม์ ผลลัพธ์ระหว่างกาลแสดงถึงผลการรับรู้ปัจจุบันสำหรับส่วนของเสียงในขณะที่ผลการจดจำขั้นสุดท้ายแสดงถึงการคาดเดาสุดท้ายที่ดีที่สุดสำหรับส่วนของเสียงนั้น

## **Streaming Speech-to-Text API Recognition Requests**

การโทรการรู้จำ Speech-to-Text แบบสตรีมมิ่งได้รับการออกแบบมาสำหรับการบันทึกและจดจำเสียงแบบเรียลไทม์ภายในสตรีมแบบสองทิศทาง แอปพลิเคชันของคุณสามารถส่งเสียงในสตรีมคำขอและรับผลการรับรู้ระหว่างกาลและขั้นสุดท้ายในสตรีมการตอบกลับแบบเรียลไทม์ ผลลัพธ์ระหว่างกาลแสดงผลการรับรู้ปัจจุบันสำหรับส่วนของเสียงในขณะที่ผลการจดจำขั้นสุดท้ายแสดงถึงการคาดเดาสุดท้ายที่ดีที่สุดสำหรับส่วนของเสียงนั้น

### **Streaming requests**

ไม่เหมือนกับการโทรแบบซิงโครนัสและแบบอะซิงโครนัสซึ่งคุณส่งทั้งการกำหนดค่าและเสียงภายในคำขอเดียวการเรียก API Speech แบบสตรีมมิงต้องส่งคำขอหลายรายการ StreamingRecognizeRequest แรกต้องมีคอนฟิกูเรชันประเภท StreamingRecognitionConfig โดยไม่มีเสียงประกอบ StreamingRecognizeRequests ที่ตามมาที่ส่งผ่านสตรีมเดียวกันจะประกอบด้วยเฟรมต่อเนื่องของไบต์เสียงดิบ

StreamingRecognitionConfig ประกอบด้วยฟิลด์ต่อไปนี้:

• config - (จำเป็น) มีข้อมูลการกำหนดค่าสำหรับเสียงประเภท RecognitionConfig และเหมือนกับที่แสดงในคำร้องขอแบบซิงโครนัสและอะซิงโครนัส

• single\_utterance - (ไม่บังคับค่าเริ่มต้นเป็นเท็จ) ระบุว่าคำขอนี้ควรสิ้นสุดโดยอัตโนมัติหรือไม่หลังจากตรวจไม่พบคำพูดอีกต่อไป หากตั้งค่าไว้ Speech-to-Text จะตรวจจับการหยุดชั่วคราวเงียบหรือเสียงที่ไม่ใช่เสียงพูดเพื่อกำหนดว่าเมื่อใดควรสิ้นสุดการจดจำ หากไม่ได้ตั้งค่าสตรีมจะรับฟังและประมวลผลเสียงต่อไปจนกว่าสตรีมจะปิดโดยตรงหรือเกินความยาวที่ จำกัด ของสตรีม การตั้งค่า single\_utterance เป็น true มีประโยชน์สำหรับการประมวลผลคำสั่งเสียง

• interim\_results - (ไม่บังคับค่าเริ่มต้นเป็นเท็จ) ระบุว่าคำขอสตรีมนี้ควรส่งคืนผลลัพธ์ชั่วคราวที่อาจได้รับการปรับแต่งในภายหลัง (หลังจากประมวลผลเสียงเพิ่มเติมแล้ว) ผลลัพธ์ระหว่างกาลจะถูกบันทึกไว้ในการตอบสนองผ่านการตั้งค่า is\_final เป็น false

### **Streaming responses**

ผลลัพธ์การรู้จำเสียงการสตรีมจะถูกส่งกลับภายในชุดของการตอบสนองประเภท StreamingRecognitionResponse คำตอบดังกล่าวประกอบด้วยฟิลด์ต่อไปนี้:

• speechEventType มีเหตุการณ์ประเภท SpeechEventType ค่าของเหตุการณ์เหล่านี้จะระบุเมื่อมีการพิจารณาว่าคำพูดเดียวเสร็จสมบูรณ์ กิจกรรมการพูดทำหน้าที่เป็นเครื่องหมายในการตอบสนองของสตรีมของคุณ

•ผลลัพธ์ประกอบด้วยรายการผลลัพธ์ซึ่งอาจเป็นผลลัพธ์ระหว่างกาลหรือผลลัพธ์สุดท้ายประเภท StreamingRecognitionResult รายการผลลัพธ์ประกอบด้วยฟิลด์ย่อยต่อไปนี้:

•ทางเลือกอื่นประกอบด้วยรายการการถอดเสียงทางเลือก

• isFinal ระบุว่าผลลัพธ์ที่ได้ในรายการนี้เป็นแบบชั่วคราวหรือถือเป็นที่สิ้นสุด

•ความเสถียรบ่งบอกถึงความผันผวนของผลลัพธ์ที่ได้รับจนถึงขณะนี้โดย 0.0 แสดงถึงความไม่เสถียรโดยสมบูรณ์ในขณะที่ 1.0 แสดงถึงเสถียรภาพที่สมบูรณ์ โปรดทราบว่าไม่เหมือนกับความเชื่อมั่นซึ่งประมาณว่าการถอดความถูกต้องเสถียรภาพจะประมาณว่าผลลัพธ์บางส่วนอาจเปลี่ยนแปลงได้หรือไม่ ถ้า isFinal ถูกตั้งค่าเป็นจริงจะไม่มีการตั้งค่าความเสถียร

* 1. Colaboratory

Colaboratory หรือเรียกสั้น ๆ ว่า Colab คือผลิตภัณฑ์จากการวิจัยของ Google Colab ช่วยให้ทุกคนสามารถเขียนและรันโค้ดหลามได้ตามอำเภอใจผ่านเบราว์เซอร์และเหมาะอย่างยิ่งกับแมชชีนเลิร์นนิงการวิเคราะห์ข้อมูลและการศึกษา ในทางเทคนิค Colab เป็นบริการโน้ตบุ๊ก Jupyter ที่โฮสต์ซึ่งไม่จำเป็นต้องตั้งค่าใด ๆ เพื่อใช้งานในขณะที่ให้การเข้าถึงทรัพยากรคอมพิวเตอร์รวมถึง GPU ได้ฟรี

2.4 งานวิจัยที่เกี่ยวข้อง (Related Works)

* **A Study on Automatic Speech Recognition**

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

* **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