Separating different speakers in an audio recording

Beta

This feature is in a pre-release state and might change or have limited support. For more information, see the product launch stages (https://cloud.google.com/products/#product-launch-stages).

This page describes how to get labels for different speakers in audio data transcribed by Speech-to-Text.

Sometimes, audio data contains samples of more than one person talking. For example, audio from a telephone call usually features voices from two or more people. A transcription of the call ideally includes who speaks at which times.

Speaker diarization

Speech-to-Text can recognize multiple speakers in the same audio clip. When you send an audio transcription request to Speech-to-Text, you can include a parameter telling Speech-to-Text to identify the different speakers in the audio sample. This feature, called *speaker diarization*, detects when speakers change and labels by number the individual voices detected in the audio.

When you enable speaker diarization in your transcription request, Speech-to-Text attempts to distinguish the different voices included in the audio sample. The transcription result tags each word with a number assigned to individual speakers. Words spoken by the same speaker bear the same number. A transcription result can include numbers up to as many speakers as Speech-to-Text can uniquely identify in the audio sample.

When you use speaker diarization, Speech-to-Text produces a running aggregate of all the results provided in the transcription. Each result includes the words from the previous result. Thus, the `words` array in the final result provides the complete, diarized results of the transcription.

Note: Review the list of supported features by language (https://cloud.google.com/speech-to-text/docs/supported-features-languages) to see the list of languages supported for this feature. Speech-to-Text only supports speaker diarization for transcribing phone calls—that is, when using the `phone_call` model (https://cloud.google.com/speech-to-text/docs/phone-model).

Enabling speaker diarization in a request

To enable speaker diarization, you need to set the `enableSpeakerDiarization` field to `true` in the `RecognitionConfig` (https://cloud.google.com/speech-to-text/docs/reference/rest/v1p1beta1/RecognitionConfig) parameters for the request. To improve your transcription results, you should also specify the number of speakers present in the audio clip by setting the `diarizationSpeakerCount` field in the `RecognitionConfig` (https://cloud.google.com/speech-to-text/docs/reference/rest/v1p1beta1/RecognitionConfig) parameters. Speech-to-Text uses a default value if you do not provide a value for `diarizationSpeakerCount`.

Cloud Speech-to-Text supports speaker diarization for all speech recognition methods: `speech:recognize` (https://cloud.google.com/speech-to-text/docs/reference/rest/v1p1beta1/speech/recognize) `speech:longrunningrecognize` (https://cloud.google.com/speech-to-text/docs/reference/rest/v1p1beta1/speech/longrunningrecognize), and Streaming (https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1p1beta1#google.cloud.speech.v1p1beta1.StreamingRecognizeRequest) .

The following code snippet demonstrates how to enable speaker diarization in a transcription request to Speech-to-Text.

PROTOCOL JAVA NODE.JS PYTHON

Refer to the `speech:recognize` (https://cloud.google.com/speech-to-text/docs/reference/rest/v1p1beta1/speech/recognize) API endpoint for complete details.

To perform synchronous speech recognition, make a `POST` request and provide the appropriate request body. The following shows an example of a `POST` request using `curl`. The example uses the access token for a service account set up for the project using the Google Cloud Cloud SDK (https://cloud.google.com/sdk). For instructions on installing the Cloud SDK, setting up a project with a service account, and obtaining an access token, see the quickstart (https://cloud.google.com/speech-to-text/docs/quickstart-protocol).

```
curl -s -H "Content-Type: application/json" \
    -H "Authorization: Bearer $(gcloud auth application-default print-access-token)
    https://speech.googleapis.com/v1p1beta1/speech:recognize \
    --data '{
    "config": {
        "encoding":"LINEAR16",
        "languageCode": "en-US",
        "enableSpeakerDiarization": true,
        "diarizationSpeakerCount": 2,
        "model": "phone_call"
    },
    "audio": {
        "uri":"gs://cloud-samples-tests/speech/commercial_mono.wav"
    }
}' > speaker-diarization.txt
```

If the request is successful, the server returns a `200 OK` HTTP status code and the response in JSON format, saved to a file named `speaker-diarization.txt`.

```
{
  "results": [
    {
      "alternatives": [
        {
          "transcript": "hi I'd like to buy a Chromecast and I was wondering whethe
          "confidence": 0.92142606,
          "words": [
            {
              "startTime": "0s",
              "endTime": "1.100s",
              "word": "hi",
              "speakerTag": 2
            },
            {
              "startTime": "1.100s",
              "endTime": "2s",
              "word": "I'd",
              "speakerTag": 2
            },
```

```
            {
              "startTime": "2s",
              "endTime": "2s",
              "word": "like",
              "speakerTag": 2
            },
            {
              "startTime": "2s",
              "endTime": "2.100s",
              "word": "to",
              "speakerTag": 2
            },
            ...
            {
              "startTime": "6.500s",
              "endTime": "6.900s",
              "word": "certainly",
              "speakerTag": 1
            },
            {
              "startTime": "6.900s",
              "endTime": "7.300s",
              "word": "which",
              "speakerTag": 1
            },
            {
              "startTime": "7.300s",
              "endTime": "7.500s",
              "word": "color",
              "speakerTag": 1
            },
            ...
          ]
        }
      ],
      "languageCode": "en-us"
    }
  ]
}
```

Last updated January 21, 2020.