Getting word timestamps

This page describes how to get time offset values for audio transcribed by Speech-to-Text.

Cloud Speech-to-Text can include *Time offset (timestamp)* (https://cloud.google.com/speech-to-text/docs/basics#time-offsets) values in the response text for your recognize request. Time offset values show the beginning and end of each spoken word that is recognized in the supplied audio. A time offset value represents the amount of time that has elapsed from the beginning of the audio, in increments of 100ms.

Time offsets are especially useful for analyzing longer audio files, where you may need to search for a particular word in the recognized text and locate it (seek) in the original audio. Cloud Speech-to-Text supports time offsets for all speech recognition methods: `speech:recognize` (https://cloud.google.com/speech-to-text/docs/reference/rest/v1/speech/recognize), `speech:longrunningrecognize` (https://cloud.google.com/speech-to-text/docs/reference/rest/v1/speech/longrunningrecognize), and `StreamingRecognizeRequest` (https://cloud.google.com/speech-to-text/docs/reference/rpc/google.cloud.speech.v1#google.cloud.speech.v1.StreamingRecognizeRequest).

Time offset values are only included for the first alternative provided in the recognition response.

To include time offsets in the results of your request, set the `enableWordTimeOffsets` parameter to `true` in your request configuration.

PROTOCOL GCLOUD COMMAND MORE

Refer to the `speech:longrunningrecognize` (https://cloud.google.com/speech-to-text/docs/reference/rest/v1/speech/longrunningrecognize) 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 -X POST \
     -H "Authorization: Bearer "$(gcloud auth application-default print-access-toke
     -H "Content-Type: application/json; charset=utf-8" \
     --data "{
  'config': {
    'language_code': 'en-US',
    'enableWordTimeOffsets': true
  },
  'audio':{
    'uri':'gs://gcs-test-data/vr.flac'
  }
}" "https://speech.googleapis.com/v1/speech:longrunningrecognize"
```

See the RecognitionConfig (https://cloud.google.com/speech-to-text/docs/reference/rest/v1/RecognitionConfig) and RecognitionAudio (https://cloud.google.com/speech-to-text/docs/reference/rest/v1/RecognitionAudio) reference documentation for more information on configuring the request body.

If the request is successful, the server returns a `200 OK` HTTP status code and the response in JSON format:

```
{
  "name": "7612202767953098924"
}
```

where `name` is the name of the long running operation created for the request.

Processing the `vr.flac` file takes about 30 seconds to complete. To retrieve the result of the operation, make a `GET` request to the `https://speech.googleapis.com/v1/operations/` endpoint. Replace ***`your-operation-name`*** with the `name` received from your `longrunningrecognize` request.

```
curl -H "Authorization: Bearer "$(gcloud auth application-default print-access-toke
     -H "Content-Type: application/json; charset=utf-8"
     "https://speech.googleapis.com/v1/operations/your-operation-name"
```

If the request is successful, the server returns a `200 OK` HTTP status code and the response in JSON format:

```
{
  "name": "7612202767953098924",
  "metadata": {
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeMetada
    "progressPercent": 100,
    "startTime": "2017-07-20T16:36:55.033650Z",
```

```
    "lastUpdateTime": "2017-07-20T16:37:17.158630Z"
  },
  "done": true,
  "response": {
    "@type": "type.googleapis.com/google.cloud.speech.v1.LongRunningRecognizeRespon
    "results": [
      {
        "alternatives": [
          {
            "transcript": "okay so what am I doing here...(etc)...",
            "confidence": 0.96596134,
            "words": [
              {
                "startTime": "1.400s",
                "endTime": "1.800s",
                "word": "okay"
              },
              {
                "startTime": "1.800s",
                "endTime": "2.300s",
                "word": "so"
              },
              {
                "startTime": "2.300s",
                "endTime": "2.400s",
                "word": "what"
              },
              {
                "startTime": "2.400s",
                "endTime": "2.600s",
                "word": "am"
              },
              {
                "startTime": "2.600s",
                "endTime": "2.600s",
                "word": "I"
              },
              {
                "startTime": "2.600s",
                "endTime": "2.700s",
                "word": "doing"
              },
              {
                "startTime": "2.700s",
                "endTime": "3s",
```

```
                "word": "here"
              },
              {
                "startTime": "3s",
                "endTime": "3.300s",
                "word": "why"
              },
              {
                "startTime": "3.300s",
                "endTime": "3.400s",
                "word": "am"
              },
              {
                "startTime": "3.400s",
                "endTime": "3.500s",
                "word": "I"
              },
              {
                "startTime": "3.500s",
                "endTime": "3.500s",
                "word": "here"
              },
              ...
            ]
          }
        ]
      },
      {
        "alternatives": [
          {
            "transcript": "so so what am I doing here...(etc)...",
            "confidence": 0.9642093,
          }
        ]
      }
    ]
  }
}
```

If the operation has not completed, you can poll the endpoint by repeatedly making the `GET` request until the `done` property of the response is `true`.

Last updated January 21, 2020.