Start Lab 01:30:00

Measuring and Improving Speech Accuracy

Lab 1 hour No cost Intermediate

GSP758

Google Cloud Self-Paced Labs

Speech Accuracy

Lab 1 hour No cost Intermediate

GSP758

Google Cloud Self-Paced Labs

3. What the Speech Adaptation API does and how it works

4. How to approach speech adaptation and language biasing on your own data

Task 1. Defining speech accuracy

Speech accuracy can be measured in a variety of ways. It may be useful for you to use multiple metrics, depending on your needs. However, the industry standard method for comparison is *word error rate*, often abbreviated as WER. Word error rate measures the percentage of incorrect transcriptions in the entire set. This means that a lower WER

means the system is more accurate.

You may also see the term, *ground truth*, used in the context of ASR accuracy. Ground truth means the 100% accurate (typically human) transcription that you will compare against to measure the accuracy.

Word Error Rate (WER)

Word error rate is the combination of three types of transcription errors which can occur:

- Insertion Error (I) - Words present in the hypothesis transcript that are not present in the ground truth
- Substitution errors (S) - Words that are present in both the hypothesis and ground truth but not transcribed correctly
- Deletion errors (D) - Words that are missing from the hypothesis but present in the ground truth

$$WER = \frac{S + D + I}{N} :$$

You combine the literal number of each one of these errors and device by the total number of words (N) in the ground truth transcript in order to find the word error rate (WER). This means that the WER can be greater than 100% in situations with very low accuracy.

Task 2. Other metrics

You may also see the use of other metrics at times, and we will demonstrate some later in the lab. These metrics can be useful for tracking things like readability or measuring how many of your most important terms were transcribed correctly. Here are a few you might encounter.

- Jaccard Index - Measures the overall similarity of the hypothesis and ground truth (or a subset of the ground truth) it is defined as the number of words that are the same over the total number of words. It can be useful to measure the Jaccard Index for a subset of particular important words that are present in the ground truth.
- F1 Score - Measures precision vs recall on a dataset. It is calculated by taking the harmonic mean of the precision and recall values calculated by comparing the hypothesis to the ground truth for a given set of words. It is useful when tuning speech systems towards specific terms to insure you are getting good recall without trading off too much precision.

Task 3. Measuring speech accuracy

Now that you know how to talk about accuracy and what to measure. These are the few simple steps you need to follow to get started on determining accuracy on your own audio.

Note: In this lab a sample audio files of ~10min is provided with associated ground truth. The steps below are not necessary to complete this lab, but will be necessary for you to measure quality on your own data.

Gather test audio files

You should gather a representative sample of the audio files for which you wish to measure quality. This sample should be random and should be as close to the target environment as possible. For example, if you want to transcribe conversations from a call center to aid in quality assurance, you should randomly select a few actual calls recorded on the same equipment that your production audio will come through, not recorded on your cell phone or computer microphone.

You will need at least 30min of audio to get a statistically significant accuracy metric. We recommend using between 30min and 3 hours of audio. In this lab the audio is provided for you.

Get ground truth transcriptions

Next you need to get accurate transcriptions of the audio. This usually involves a single or double pass human transcription of the target audio. Your goal is to have a 100% accurate transcription to measure the automated results against.

It's important when doing this to match the transcription conventions of your target ASR system as closely as possible. For example, ensure that punctuation, numbers, and capitalization are consistent. In this lab the ground truth is provided for you.

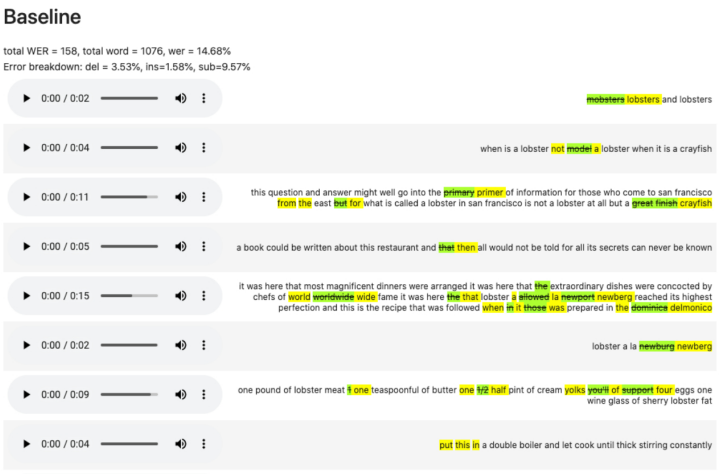
Get the machine transcription

Send the audio to Google Speech-to-Text API and get your hypothesis transcription. You can do this using one of Google Cloud's many libraries or command line tools. In this lab the code to do this is provided for you.

Compute the WER

Now you must count the insertions, substitutions, deletions, and total words. You can do this using the ground truth from step 2 and machine transcription from step 3.

Google has created and opened source the code used to normalize output and calculate the WER for you.



In this lab code to do this is provided for you.

Task 4. Try measuring accuracy

Now you will try this out for real. This lab has curated and created a focused dataset based on public domain books and audio from the Librispeech project. All the code you need to measure the accuracy of Google Cloud Speech-to-Text API's accuracy on this dataset is provided.

In the following Notebook you will learn how to set up and use this code. Once you have launched the notebook, follow the instructions inside to compute the WER on the provided dataset.

Create the Notebook instance

To create and launch a Vertex AI Workbench notebook:

1. In the **Navigation Menu** ☰, click **Vertex AI** > **Workbench**.
2. On the **Workbench** page, click **Enable Notebooks API** (if it isn't enabled yet).
3. Click on **User-Managed Notebooks** tab then, click **Create New**.
4. Name the notebook.
5. Set **Region** to `REGION` and **Zone** to `ZONE`.
6. In the **New instance** menu, choose the latest version of **TensorFlow Enterprise 2.11** in **Environment**.
7. Click **Advanced Options** to edit the instance properties.
8. Click **Machine type** and then select **e2-standard-2** for Machine type.
9. Leave the remaining fields at their default and click **Create**.

After a few minutes, the **Workbench** page lists your instance, followed by **Open JupyterLab**.

10. Click **Open JupyterLab** to open JupyterLab in a new tab. If you get a message saying beatrix jupyterlab needs to be included in the build, just ignore it.

Load the notebook

1. Under **Other**, click **Terminal**.
2. Run the following commands to copy the notebooks you will work with:

```
gsutil cp gs://spls/gsp758/notebook/measuring-accuracy.ipynb .
```

```
gsutil cp gs://spls/gsp758/notebook/speech_adaptation.ipynb .
```

```
gsutil cp gs://spls/gsp758/notebook/simple_wer_v2.py .
```

Perform the following tasks to Play Audio Files in an Incognito Window:

1. Within Chrome click on the **3 dots** > **Settings**.
2. In the **Search Settings** type "Incognito".
3. In the results, click on **Third-party cookies**.
4. Go to **Allowed to use third-party cookies**.
5. Click **Add**.

6. Copy the JUPYTERLAB domain, do not include https.

It should be something like:

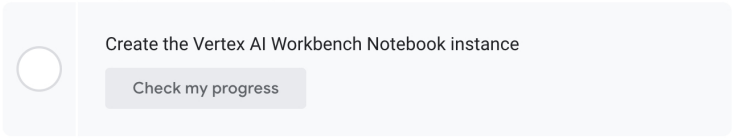
```
[YOUR_NOTEBOOK_ID].notebooks.googleusercontent.com
```

7. check **Current incognito session only** click **add**.

You can now continue to the notebook.

8. Open the **measuring-accuracy.ipynb** notebook to follow the instructions inside to compute the WER on the provided dataset.

Click *Check my progress* to verify the objective.



Task 5. Improving speech accuracy

Now that you have measured the accuracy of Google Cloud Speech-to-Text on your provided dataset, it's time to start thinking about how you can improve on the results you already have.

There are many ways to think about how you can give the ASR system more signal to improve the accuracy and lower the WER. The following three are some things to consider as you get started.

1. Customize the model to your domain by providing contextual information.

- e.g. Say you are creating a bot that allows people to order pizza. You might want to increase the probability that words like pepperoni, olives, and mozzarella are recognized.

2. Tweak weights to address specific word / phrase issues.

- e.g. Say you are trying to recognize proper nouns, rare words, or made up words. It's unlikely that these will be transcribed correctly initially, biasing towards them can fix individual terms.

3. Use context to bias towards specific types of information or words.

- e.g. Say you have an IVR telephone system and have just asked someone for their order number. You can bias specifically towards an alphanumeric entry.

When evaluating quality, look at where the system makes errors. You should think about if any of the above three types of context could help give the system more signal and improve accuracy.

If you think you can provide this type of context and get an improvement, you can do it with the Speech Adaptation API available in the Cloud Speech-to-Text API.

Task 6. Speech adaptation

Google Cloud Speech-to-Text has tools for providing contextual information that can help users increase accuracy on their data. The Speech Adaptation API allows users to pass phrases and associated weights directly to the speech API.

These phrases can be changed with every request and allow for quick iteration as well as on the fly adaptation. All you do is include the terms in the request itself as part of the recognition config:

```
"speech_contexts": [{
    "phrases": ["foo", “bar”],
    "boost": 10.0
  },{
    "phrases": ["foo bar", “bar foo”],
    "boost": 5.0
  }]
}
```

This type of biasing is advantageous over methods such as custom language models or complex grammars. It is easier to setup, doesn’t require any special training or deployment, and is included for free in your usage of the Cloud Speech-to-Text API.

Task 7. Determining speech adaptation terms

As you learned above, Cloud Speech-to-Text makes it very easy to bias the system. However, you still have to figure out the right terms to send to the API. Thinking back to the previous considerations about types of quality improvements, you can also consider the following when deciding what terms to include with biasing.

- What am I doing with this transcript? - Is there a downstream system that will be sensitive to particular words or phrases?
    - These words or phrases should be bias towards since getting them correct is very important.
- Are there rare words or proper nouns?
    - These words or phrases may not be predicted correctly since they occur infrequently and should be biased towards.
- What contextual info can I use? - Do you know what words somebody might say or what they said in the past?
    - These can be biased towards to help increase accuracy even on commonly occurring words if you are sure they will be present
- Do you have “strong” or “weak” context?
    - You can bias heavily with “strong” context if you are sure the user is about to mention some specific words
    - You should bias less if you have “weak” context meaning you know what words will occur but not exactly sure where or when.

Task 8. Try improving accuracy with speech adaptation

Now that you have learned some about how to approach biasing, it's time to put it into practice on the dataset from before. However, the performance on our old dataset was already pretty good. To make the problem a little harder for the ASR system, noise has been added to the audios that were provided in the previous notebook.

- Launch the **speech_adaptation.ipynb** notebook and follow the steps to check the accuracy on the noisy file, then try out using the Speech Adaptation API to iterate on potential phrase and boost configurations. Finally, the configuration thought to fit best is provided.

Congratulations

You have now successfully measured and improved the accuracy of Google Cloud Speech-to-Text API on a real dataset. You have learned how to talk about and compare accuracy metrics and how to approach measuring accuracy. You have successfully set up Python tools for performing automated speech recognition and measuring accuracy.

You are now ready to try these tools on your own data and put what you have learned into practice.

Finish your quest

This self-paced lab is part of the Language, Speech, Text & Translation with Google CLoud APIs quest. A quest is a series of related labs that form a learning path. Completing this quest earns you a badge to recognize your achievement. You can make your badge or badges public and link to them in your online resume or social media account. Enroll in the above quest and get immediate completion credit. Refer to the Google Cloud Skills Boost catalog for all available quests.

Take your next lab

Continue your quest with Translate Text with the Cloud Translation API or try one of these:

- Classify Text into Categories with the Natural Language API
- Entity and Sentiment Analysis with the Natural Language API

Google Cloud training and certification

...helps you make the most of Google Cloud technologies. Our classes include technical skills and best practices to help you get up to speed quickly and continue your learning journey. We offer fundamental to advanced level training, with on-demand, live, and virtual options to suit your busy schedule. Certifications help you validate and prove your skill and expertise in Google Cloud technologies.

Manual Last Updated October 25, 2023

Manual Last Tested October 25, 2023