Group 8 : Ananya Bhanja Chowdhury (385932)
Aslihan Özmen (387980)
Navya Basavaraju(386099)
Yuchun Chen(387275)

Google Cloud Speech API Benchmark Design

1. Introduction (Ananya)

The goal of this project is to benchmark the Google Cloud Speech API and evaluate the results. Google Cloud Speech API[1] is a powerful automated speech recognition (ASR) system that uses neural network models to process hundreds of languages and accents from across the world. It supports WAV and FLAC audio format. It allows users to transcribe files stored in Google Cloud Storage (bucket) or by sending the audio file in the API request.

2. Benchmarking Goals (Yuchun)

Our benchmarking goal is to evaluate the completeness and correctness of the audio files transcribed by the Google Cloud Speech API. We have analyzed the Google Cloud Speech API output with respect to reference files. We have classified metrics with different language factor (English and French) and different sample rate. Moreover, we would like to determine how serious the background noise impact the evaluation result with same dataset.

3. Benchmarking Tools (All)

Kaldi[2] is an open source toolkit commonly used for speech recognition.

- **Structure**

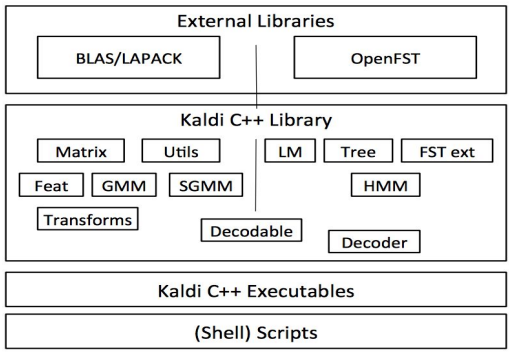


Figure 1 Internal structure of Kaldi[3]

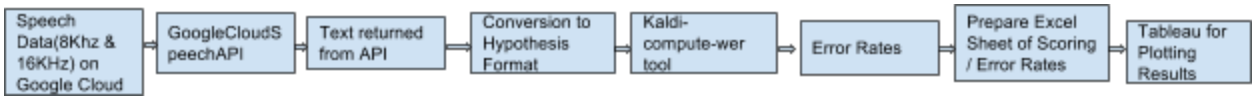
- **Benefit**

First of all, Kaldi is open source for ASR system. It's extendable and redistributable and could apply to benchmark the google cloud speech API. Also, there are few online documentation described how we could setup Kaldi which made the environment setup easy. Moreover, it integrated with OpenFst and make the computation efficient.

---

[1] Available from https://cloud.google.com/speech/
[2] Available from http://kaldi-asr.org/doc/tools.html
[3] [2] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, and J. Silovsky, "The Kaldi speech recognition toolkit," in IEEE 2011 Workshop on Automatic Speech Recognition and Understanding (No. EPFL-CONF- 192584), IEEE Signal Processing Society, 2011.

4. Experiment Setup and System / Resource Configurations



- **Data (Yuchun & Aslihan)**

We use the dataset from the VoxForge corpus[4] and University of Edinburgh[5].

VoxForge is the collection of transcribed speech for several languages and also the standard dataset highly referenced in research papers for speech recognition. We have chosen English and French languages for our experiment, 50 files each one from 4 to 7 seconds for both language and total 100 audio samples.

We obtained the dataset from University of Edinburgh in order to verify the WER with/without noise.

The data could be mainly classified with four characteristics viz., sample rate (E.g. 16000 Hz/ 8000 Hz) , data size (E.g. 8/16 bits), data encoding (E.g. u-law,a-law), and channels (E.g. Stereo 2 channels). But in this benchmarking design, we are concerned about the sample rate only and all audios are formatted as 8kHz and 16kHz WAV files separately. Additionally, our sampling data has different dialects as well, for example Indian accent english, American English, French of France ,and French of Africa and so on.

Sound eXchange (SoX)[6] is the sound processing program which we use to convert various formats of audio files into WAV format. In addition, we convert the sample rate to required output (8kHz and 16kHz) through SoX as well.

- **Benchmarking Tools Setup (Navya)**
  - **Installation : Kaldi- Speech Recognition Toolkit**

The compiled Kaldi will take up to 15 gigs of disk space, so it is important to allocate it on the instance during storage setup. We setup Kaldi in a VM on Google Cloud with **Persistent Storage of 30GB and vCPU as 1.**

Boot disk and local disks

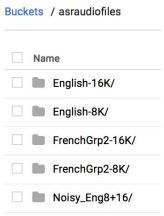
Name	Size (GB)	Type	Mode
instance-1	30	Standard persistent disk	Boot, read/write

```
g++  -Wl,-rpath=/home/nbmastersde/kaldi/tools/openfst/lib  -rdynamic
.so  ../nnet3/libkaldi-nnet3.so  ../cudamatrix/libkaldi-cudamatrix.so
transform/libkaldi-transform.so  ../gmm/libkaldi-gmm.so  ../tree/libka
ldi/tools/openfst/lib/libfst.so /usr/lib/libatlas.so.3 /usr/lib/libf77
make[2]: Leaving directory '/home/nbmastersde/kaldi/src/rnnlmbin'
make[1]: Leaving directory '/home/nbmastersde/kaldi/src'
echo Done
Done
nbmastersde@instance-1:~/kaldi/src$ cd ..
nbmastersde@instance-1:~/kaldi$ ls
COPYING  egs  INSTALL  misc  README.md  scripts  src  tools  windows
```

[4] Available from http://www.voxforge.org/home/downloads
[5] Available from https://datashare.is.ed.ac.uk/handle/10283/2791
[6] Available from http://sox.sourceforge.net

- **Resource configurations (Ananya)**

Uploaded speech to be recognized as **publicly available audio files of .wav format on google cloud** in a bucket[7]



Further setup and experiment info can be found on public Git-repository[8] on GitLab TU Berlin

- **Cleanup Phase**

After experiment, we will have to delete the objects(Audio files) from the bucket.

5. Result (All)

We are calculating the Word Error Rate (WER) and exact match of the files transcribed using Google Speech API. In this section we briefly discuss the concepts of WER, Exact match and present our results.

- **Word Error Rate (Correctness)**

In the evaluation result, WER means “Word Error Rate” and the formula is as below. It compares a reference (original text) to a hypothesis(result formatted for Kaldi) and the metric describes the percentage of errors on word recognition.

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

where...
S = number of substitutions
D = number of deletions
I = number of insertions
N = number of words in the reference

Figure 2 The Formula of WER Calculation [9]

The errors are including the cases of substitutions, deletions and insertions of characters. You could refer to Table 1 example in details. Apart from that, the uppercase and lowercase of characters would consider as substitution as well.

Table 1. Examples of Substitution, Insertion, and Deletion

Expected Output	This is **a** book
Substitution	This is **an** book
Insertion	This is a **new** book
Deletion	This is book

[7] https://goo.gl/B6UJj5
[8] https://gitlab.tubit.tu-berlin.de/leonav/EC-GoogleSpeechAPI-Benchmarking
[9] Available from https://sonix.ai/articles/word-error-rate

SER means “Sentence Error Rate” and the comparison between reference transcript and hypothesis transcript and it seems much more restrict as the entire sentence should be 100% match, we could see from the code “sent_errs += (ref_sent != hyp_sent);”[10] as shown.
The performance result is shown in Table 2 and the plot for the result are available in the poster.
The Benchmark execution screenshots are available in gitlab repository.

Table 2. The Result of Word Error Rate for ENG/FR with Different Sample Rate

	WER 8KHz	WER 16KHz	SER 8KHz	SER 16KHz
English	17.13	14.71	78	60
French	21.87	17.99	84	80

- **Exact match, Detection Rate (Completeness)**

In order to determine the detection rate clearly. We find the scenarios with noisy background and clean background to verify the influence of WER result. The performance result is shown in Table 3 and the plot for the result are available in the poster.

Table 3. The Result of Word Error Rate for Noiseless/Noisy Scenario with Different Sample Rate

	WER 8KHz	WER 16KHz	SER 8KHz	SER 16KHz
English Noiseless	24.58	16.1	80	70
English Noisy	10.26	9.4	40	40

6. Conclusion (Aslihan)

As you can see from the experiment results, when we do comparison among English and French audio files, word and sentence error rates are much higher for French language compared to English. If there is a background noise in the audio files then the word and sentence error rates much higher. Also, we used 100 audio sample files for our benchmarking with sampling rates between 8KHz and 16KHz. It would be interesting to run the same benchmark test with a larger sample size and of higher sampling rates (~> 16KHz).

[10] Code is available from kaldi/src/bin/compute-wer.cc