MedASR Model Card

Model documentation: MedASR

(https://developers.google.com/health-ai-developer-foundations/medasr)

Resources:

- Model on Google Cloud Model Garden: MedASR (https://console.cloud.google.com/vertex-ai/publishers/google/model-garden/medasr)
- Model on Hugging Face: MedASR (https://huggingface.co/google/medasr)
- GitHub repository (supporting code, Colab notebooks, discussions, and issues): MedASR (https://github.com/google-health/medasr)
- Quick start notebook: GitHub (https://github.com/google-health/medasr/blob/main/notebooks/quick_start_with_hugging_face.ipynb)
- Fine-tuning notebook: GitHub (https://github.com/google-health/medasr/blob/main/notebooks/fine_tune_with_hugging_face.ipynb)
- Support: See Contact (https://developers.google.com/health-ai-developer-foundations/medasr/get-started.md#contact)
- License: The use of MedASR is governed by the Health AI Developer Foundations terms o (https://developers.google.com/health-ai-developer-foundations/terms).

Author: Google

Model information

This section describes the MedASR (Medical Automated Speech Recognition) model and how to use it.

Description

MedASR is a speech-to-text model based on the Conformer architecture (https://arxiv.org/abs/2005.08100) pre-trained for medical dictation. MedASR is intended as a starting point for developers, and is well-suited for dictation tasks involving medical terminologies, such as

radiology dictation. While MedASR has been extensively pre-trained on a corpus of medical audio data, it may occasionally exhibit performance variability when encountering terms outside of its pre-training data, such as non-standard medication names or consistent handling of temporal data (dates, times, or durations).

How to use

The following are some example code snippets to help you quickly get started running the model locally. If you want to use the model at scale, we recommend that you create a production version using Model Garden (https://console.cloud.google.com/vertex-ai/publishers/google/model-garden/medasr).

First, install the Transformers library. MedASR is supported starting from transformers 5.0.0. You may need to install transformers from GitHub.

```
$ uv pip install git+https://github.com/huggingface/transformers.git@65dc261512cbdb1e
```

Run model with the pipeline API

```
from transformers import pipeline
import huggingface_hub
from IPython.display import Audio, display
audio = huggingface_hub.hf_hub_download('google/medasr', 'test_audio.wav')
model_id = "google/medasr"
pipe = pipeline("automatic-speech-recognition", model=model_id)
result = pipe(audio,chunk_length_s=20,stride_length_s=2)
# the chunk length is how long in seconds MedASR batches audio and the stride le
print(result)
```


Run the model directly

```
from transformers import AutoModelForCTC, AutoProcessor
import huggingface_hub
import librosa
import torch
audio = huggingface_hub.hf_hub_download('google/medasr', 'test_audio.wav')
```

```
model_id = f"google/medasr"
device = "cuda" if torch.cuda.is_available() else "cpu"
processor = AutoProcessor.from_pretrained(model_id)
model = AutoModelForCTC.from_pretrained(model_id).to(device)
audio = huggingface_hub.hf_hub_download('google/medasr', 'test_audio.wav')
speech, sample_rate = librosa.load(audio, sr=16000)
inputs = processor(speech, sampling_rate=sample_rate, return_tensors="pt", padding=Tr
inputs = inputs.to(device)
outputs = model.generate(**inputs)
decoded_text = processor.batch_decode(outputs)[0]
print(f"result={decoded_text}")
```

✦ Code Tutor

Examples

See the following tutorial notebooks for examples of how to use MedASR:

- To give the model a quick try, running it locally with weights from Hugging Face, see Quick start notebook in Colab (https://colab.research.google.com/github/google-health/medasr/blob/main/notebooks/quick_start_with_hugging_face.ipynb) .
- For an example of fine-tuning the, see the Fine-tuning notebook in Colab (https://colab.research.google.com/github/google-health/medasr/blob/main/notebooks/fine_tune_with_hugging_face.ipynb) .

Can I help?

Model architecture overview

The MedASR model is built based on the Conformer (https://arxiv.org/abs/2005.08100) architecture.

Technical specifications

- **Model type**: Automated-speech-detector
- **Input Modalities**: Mono-channel audio 16kHz, int16 waveform
- **Output Modality:** Text only
- **Number of parameters:** 105M

- **Key publication**: LAST: Scalable Lattice-Based Speech Modelling in JAX (https://arxiv.org/pdf/2304.13134)
- **Model created**: December 18, 2025
- **Model version**: 1.0.0

Citation

When using this model, cite:
@inproceedings{wu2023last,
title={Last: Scalable Lattice-Based Speech Modelling in Jax},
author={Wu, Ke and Variani, Ehsan and Bagby, Tom and Riley, Michael},
booktitle={ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)},
pages={1--5},
year={2023},
organization={IEEE}
}

Performance and Evaluations

Our evaluation methods include evaluating word-error rate (WER) of MedASR against held out medical audio examples. We also evaluate specifically medical WER, where we only look at words that have a medical context. These audio samples have been transcribed by human experts, but there is always some noise in such transcriptions.

Key performance metrics

Word error rate of MedASR versus other models*

Dataset name	Dataset description	MedASR with greedy decoding	MedASR + 6-gram language model	Gemini 2.5 Pro	Gemini 2.5 Flash	Whisper v3 Large
RAD-DICT	Private radiologist dictation dataset	6.6%	**4.6%**	10.0%	24.4%	25.3%
GENERAL-DICT	Private general and internal medicine dataset	9.3%	**6.9%**	16.4%	27.1%	33.1%

Dataset name	Dataset description	MedASR with greedy decoding	MedASR + 6-gram language model	Gemini 2.5 Pro	Gemini 2.5 Flash	Whisper v3 Large
FM-DICT	Private family medicine dataset	8.1%	**5.8%**	14.6%	19.9%	32.5%
Eye Gaze (https://physionet.org/content/egd-cxr/1.0.0/)	Dictation of audio from 998 MIMIC cases	6.6%	**5.2%**	5.9%	9.3%	12.5%

*All results except "MedASR + 6-gram language model" in the preceding table use greedy decoding. "MedASR + 6-gram language model" uses beam search with beam size 8.

Safety evaluation

Our evaluation methods include structured evaluations and internal red-teaming testing of relevant safety policies. This model was evaluated across various dimensions to assess safety. Human evaluations were conducted on 100 example outputs to assess for potential safety impact, specifically related to incorrect transcriptions associated with medication names, dosages, diagnoses, semantic changes, and medical terminology. The results of these evaluations were determined to be acceptable in regards to internal policies for overall safety.

Data card

Dataset overview

Training

The MedASR model is specifically trained on a diverse set of de-identified medical speech data. Its training utilizes approximately 5000 hours of physician dictations across a range of specialities (proprietary dataset). The model is trained on audio segments paired with corresponding transcripts and metadata, also including extensive annotations for medical named entities such as symptoms, medications, and conditions. MedASR therefore has a strong understanding of vocabulary used in medical contexts.

Evaluation

MedASR has been evaluated using a mix of internal and public datasets as noted in the Key Performance Metrics section. We used argmax of the model for posterior probability (greedy

decoding) to get the output model's hypothesis tokens. The hypothesis is compared against ground truth transcript using jiwer library to calculate the word error rate.

Source

The datasets used to train MedASR include a public dataset for pre-training and a proprietary dataset that was licensed and incorporated (described in the following section).

Data ownership and documentation

Pre-training with the full LibriHeavy training set. (https://arxiv.org/abs/2309.08105) Fine-tuning was conducted on de-identified, licensed datasets described in the following section

Private Medical Dict: Google internal dataset consisting of de-identified dictations made by physicians of different specialities including radiology, internal medicine, family medicine, and other subspecialties totaling more than 5000 hours of audio. This dataset was split into test sets that constitute RAD-DICT, FM-DICT and General and Internal Medicine-DICT referenced previously in Performance and Evaluations.

Data citation

Eye Gaze Data for Chest X-rays (evaluation set described previously in Performance and Evaluations) was derived from:

MIMIC-CXR Database v1.0.0 and MIMIC-IV v0.4

De-identification/anonymization:

Google and its partners utilize datasets that have been rigorously anonymized or de-identified to ensure the protection of individual research participants and patient privacy.

Implementation Information

Details about the model internals.

Hardware

Tensor Processing Unit (TPU) (https://cloud.google.com/tpu/docs/intro-to-tpu) hardware (TPUv4p, TPUv5p and TPUv5e). Training speech-to text models requires significant computational power. TPUs, designed specifically for matrix operations common in machine learning, offer several advantages in this domain:

- Performance: TPUs are specifically designed to handle the massive computations involved in training VLMs. They can speed up training considerably compared to CPUs.
- Memory: TPUs often come with large amounts of high-bandwidth memory, allowing for the handling of large models and batch sizes during training. This can lead to better model quality.
- Scalability: TPU Pods (large clusters of TPUs) provide a scalable solution for handling the growing complexity of large foundation models. You can distribute training across multiple TPU devices for faster and more efficient processing.
- Cost-effectiveness: In many scenarios, TPUs can provide a more cost-effective solution for training large models compared to CPU-based infrastructure, especially when considering the time and resources saved due to faster training.
- These advantages are aligned with Google's commitments to operate sustainably (https://sustainability.google/operating-sustainably/).

Software

Training was done using JAX (https://github.com/jax-ml/jax) and ML Pathways (https://blog.google/technology/ai/introducing-pathways-next-generation-ai-architecture/). JAX allows researchers to take advantage of the latest generation of hardware, including TPUs, for faster a more efficient training of large models. ML Pathways is Google's latest effort to build artificially intelligent systems capable of generalizing across multiple tasks. This is specially suitable for foundation models, including large language models like these ones.

Together, JAX and ML Pathways are used as described in the paper about the Gemini family of models (https://goo.gle/gemma2report); *"the 'single controller' programming model of JAX and Pathways allows a single Python process to orchestrate the entire training run, dramatically simplifying the development workflow."*

Usage and Limitations

The MedASR model has certain limitations that users should be aware of.

Intended Use

MedASR is a speech-to-text model intended to be used as a starting point that enables more efficient development of downstream healthcare applications requiring speech as input. MedASR is intended for developers in the healthcare and life sciences space. Developers are responsible for training, adapting, and making meaningful changes to MedASR to accomplish their specific intended use. The MedASR model can be fine-tuned by developers using their own proprietary data for their specific tasks or solutions.

MedASR is trained on many medical audio, speech, and text and enables further development and integration, or both with generative models like MedGemma (https://developers.google.com/health-ai-developer-foundations/medgemma), where MedASR converts speech to text, which can then be used as input for a text-to-text response. Full details of all the tasks MedASR has been evaluated and pre-trained on can be found in the MedASR model card.

MedASR is not intended to be used without appropriate validation, adaptation, or making meaningful modification by developers for their specific use case. The outputs generated by MedASR may include transcription errors and are not intended to directly inform clinical diagnosis, patient management decisions, treatment recommendations, or any other direct clinical practice applications. All outputs from MedASR should be considered preliminary and require independent verification, clinical correlation, and further investigation through established research and development methodologies.

Limitations

- Training Data
    - English-only: All training data is in English
    - Speaker diversity: Most training data comes from speakers where English is their first language and were raised in the United States. The base model's performance may be lower for other types of speakers, necessitating the need for fine-tuning.
    - Speaker Sex/Gender: Training data included both men and women but had a higher proportion of men.
    - Audio quality: Training data is mostly from high quality microphones. The base model's performance may deteriorate on low quality audio with background noise, necessitating the need for fine-tuning.
    - Specialized medical terminology: Although MedASR has specialized medical audio training, its training may not include all medications, procedures or terminology, especially ones that have come into usage in the past 10 years.

- Dates: MedASR has been trained on de-identified data so its performance on different date formats may be lacking. This can be rectified with further finetuning or alternative decoding approaches such as language model decoding debiasing.

Benefits

At the time of release, MedASR is a high performing open speech-to-text model, with specific training for medical applications. Users can update its vocabulary with few-shot fine-tuning or decoding with external language models.

Based on the benchmark evaluation metrics in this document, MedASR represents a significant leap forward in medical speech-to-text performance relative to other comparably-sized open model alternatives.