WHISPERING INTO THE FUTURE

Reinventing Speech-to-Text Transcriptions with Go and Whisper

PRESENTED TO
Conf42

PRESENTED BY
Pratim Bhosale

Powered by
SurrealDB

Pratim Bhosale

Developer Advocate

Backend Developer

Technical Writer

Content

- ✓ Speech to Text transcription APIs are expensive
- ✓ What is Whisper & Whisper.cpp?
- ✓ What are Go bindings?!
- ✓ Transcription Usecases
- ✓ Demo

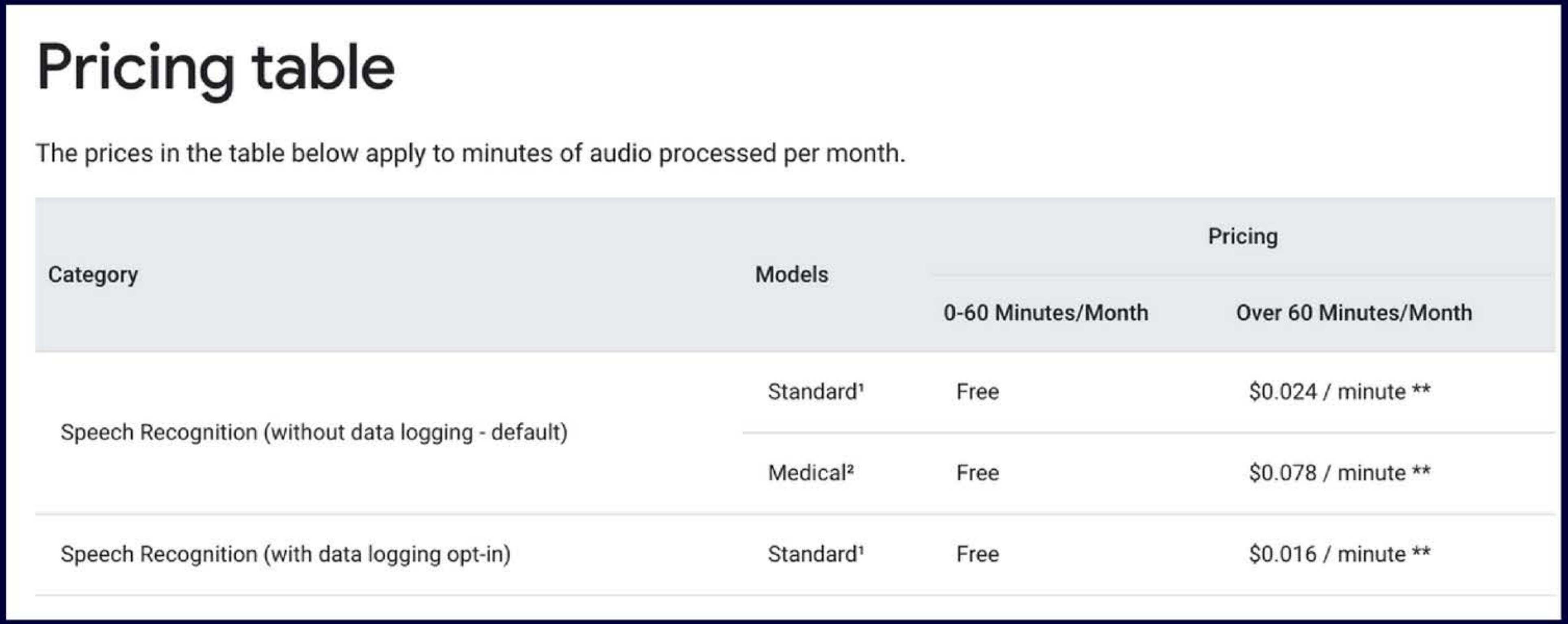
@BhosalePratim

Speech to Text transcription APIs are expensive

WHEN YOU REALIZE SPEECH TO TEXT
APIS CLEANED OUT YOUR BANK ACCOUNT
Supermeme.ai

@BhosalePratim

@BhosalePratim

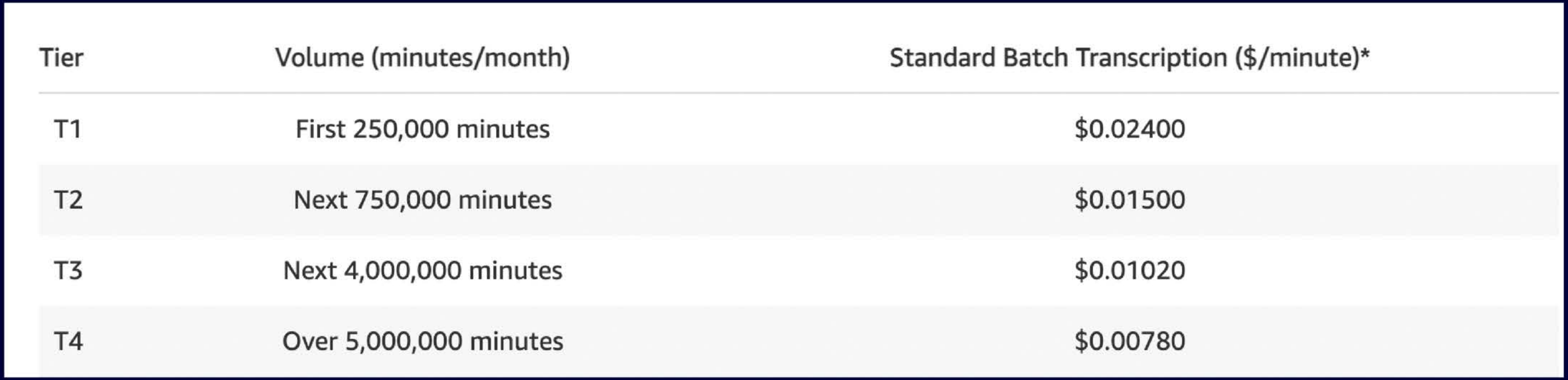


Pricing table

The prices in the table below apply to minutes of audio processed per month.

Category	Models	Pricing	
		0-60 Minutes/Month	Over 60 Minutes/Month
Speech Recognition (without data logging - default)	Standard[1]	Free	$0.024 / minute **
	Medical[2]	Free	$0.078 / minute **
Speech Recognition (with data logging opt-in)	Standard[1]	Free	$0.016 / minute **

Google's Speech-to-Text API pricing

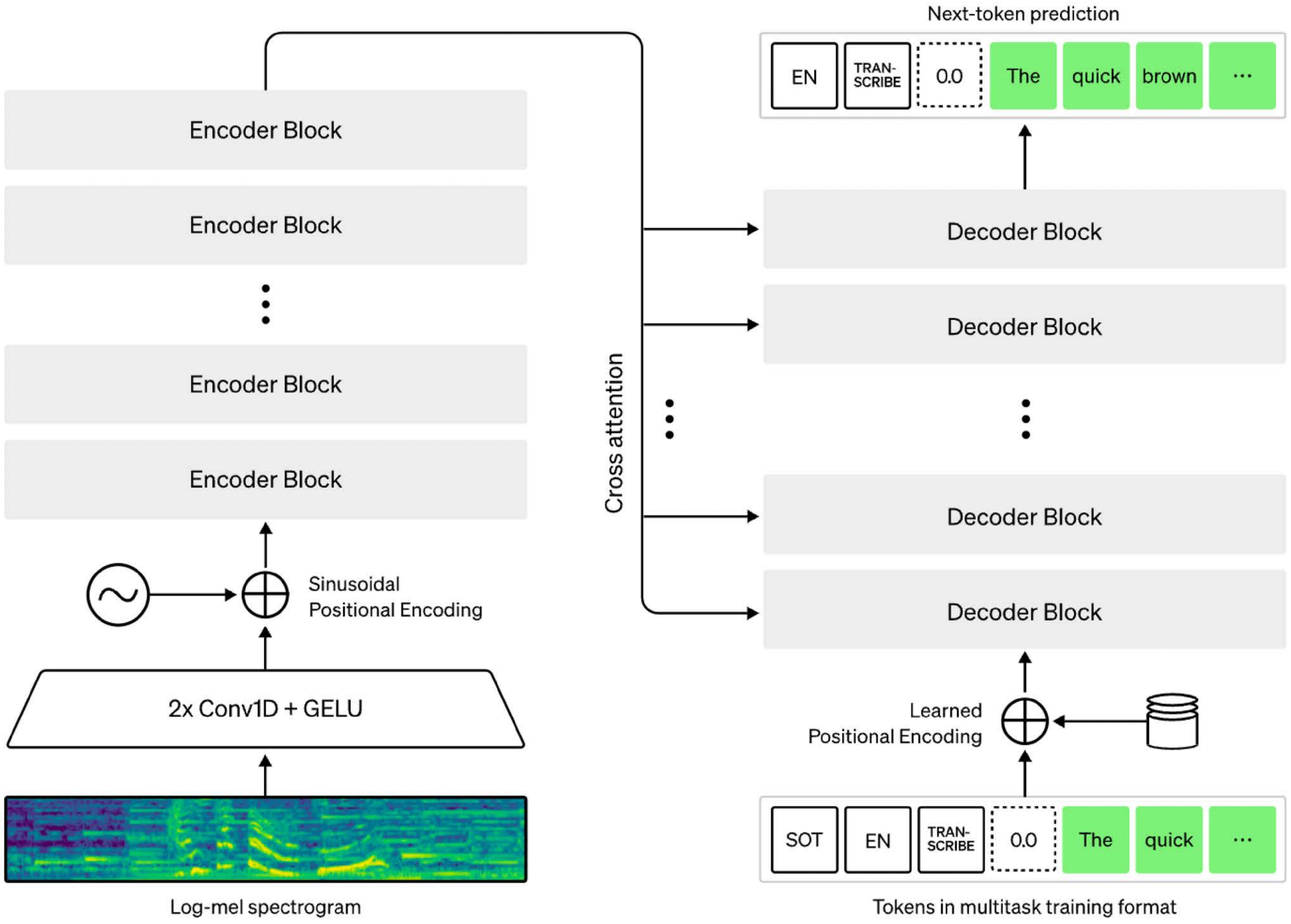


Tier	Volume (minutes/month)	Standard Batch Transcription ($/minute)*
T1	First 250,000 minutes	$0.02400
T2	Next 750,000 minutes	$0.01500
T3	Next 4,000,000 minutes	$0.01020
T4	Over 5,000,000 minutes	$0.00780

Amazon's speech to text api cost

What is Whisper and Whisper.cpp

Whisper is the most underrated OpenAI model


Next-token prediction
EN
TRAN-SCRIBE
0.0
The
quick
brown
...
Encoder Block
Encoder Block
Encoder Block
Encoder Block
Cross attention
Decoder Block
Decoder Block
Decoder Block
Decoder Block
Sinusoidal Positional Encoding
2x Conv1D + GELU
Log-mel spectrogram
Learned Positional Encoding
SOT
EN
TRAN-SCRIBE
0.0
The
quick
...
Tokens in multitask training format

- A lightweight implementation of OpenAI's Whisper speech-to-text model
- Compatible with the Go stack, thanks to its Go bindings.
- It's a cost-effective alternative to Google, Amazon and IBM APIs.
- It's Open Source.
- Can be embedded.

What are Go bindings?

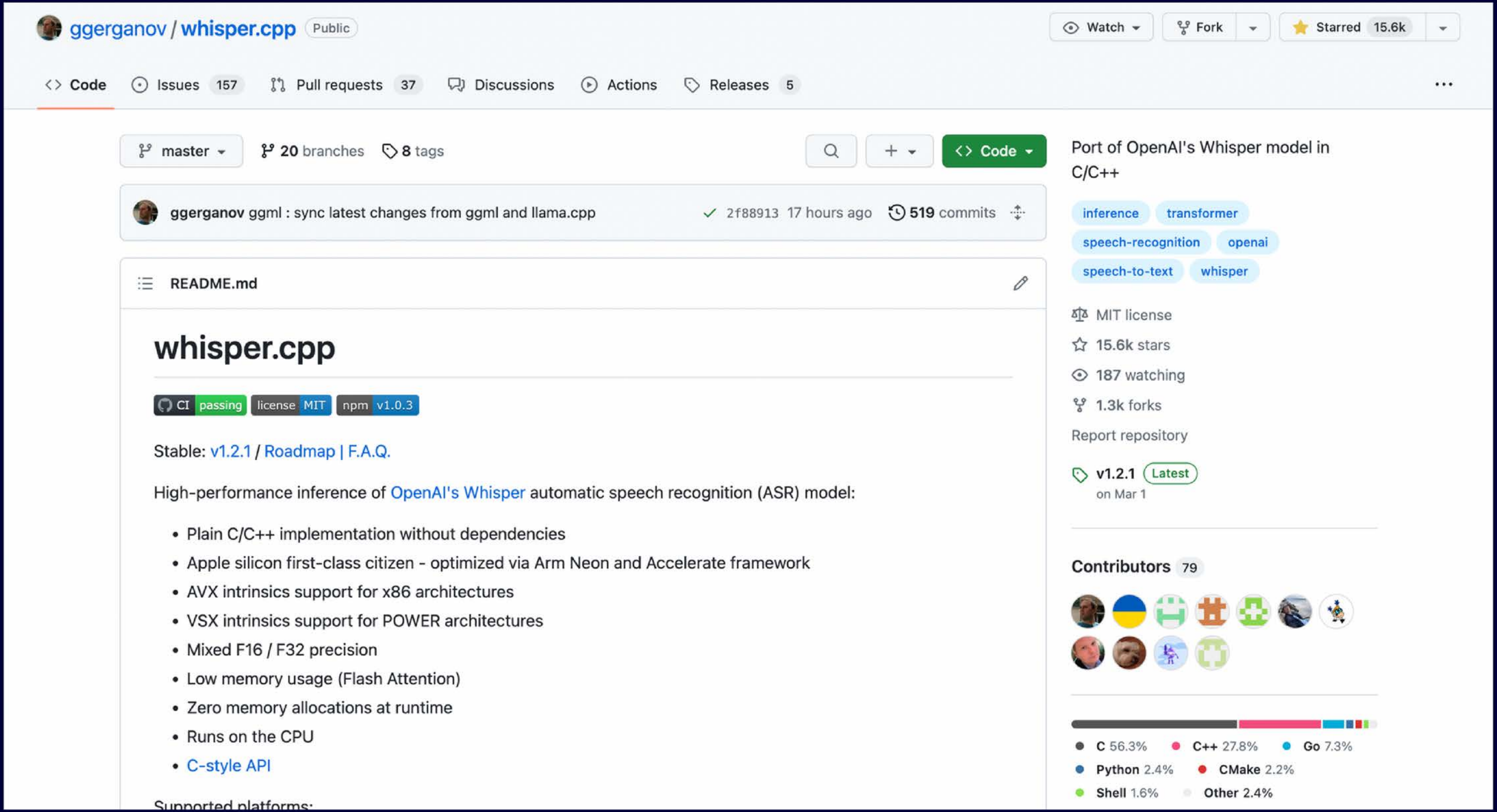
@BhosalePratim

- Go bindings are a way to interface and interact with a library or a package is written in another programming language (such as C or C++) from within Go code.

- They serve as a bridge between the Go code and the foreign language code, allowing the two to communicate and share data with each other.

- The whisper.cpp library takes care of the Go bindings.

Getting started with Whisper.cpp



ggerganov / whisper.cpp Public

Watch · Fork · Starred 15.6k

Code · Issues 157 · Pull requests 37 · Discussions · Actions · Releases 5

master · 20 branches · 8 tags · Code

ggerganov ggml : sync latest changes from ggml and llama.cpp ✓ 2f88913 17 hours ago · 519 commits

README.md

whisper.cpp

CI passing · license MIT · npm v1.0.3

Stable: v1.2.1 / Roadmap | F.A.Q.

High-performance inference of OpenAI's Whisper automatic speech recognition (ASR) model:

- Plain C/C++ implementation without dependencies
- Apple silicon first-class citizen - optimized via Arm Neon and Accelerate framework
- AVX intrinsics support for x86 architectures
- VSX intrinsics support for POWER architectures
- Mixed F16 / F32 precision
- Low memory usage (Flash Attention)
- Zero memory allocations at runtime
- Runs on the CPU
- C-style API

Supported platforms:

Port of OpenAI's Whisper model in C/C++

inference · transformer · speech-recognition · openai · speech-to-text · whisper

MIT license

15.6k stars

187 watching

1.3k forks

Report repository

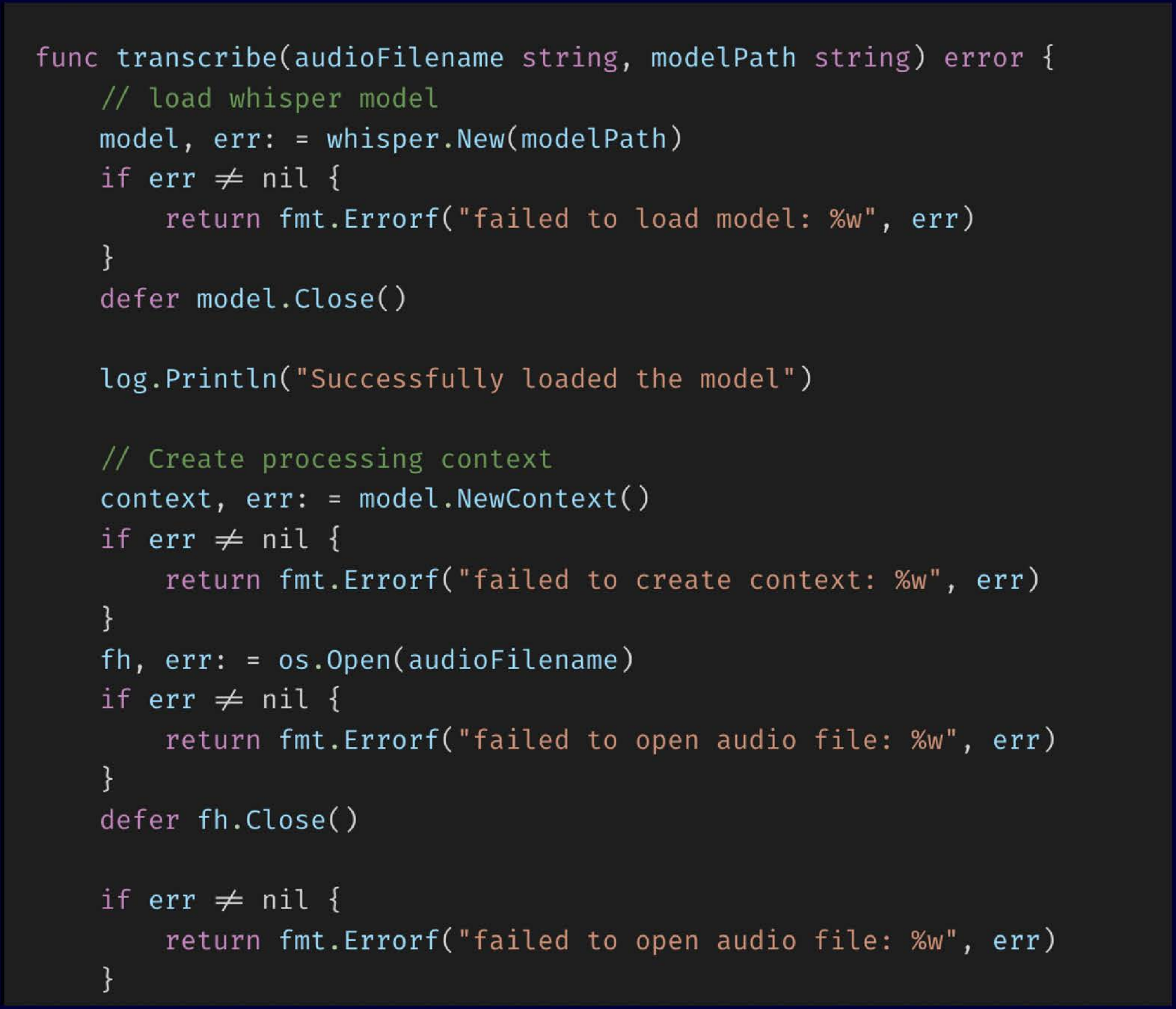
v1.2.1 Latest on Mar 1

Contributors 79

● C 56.3% ● C++ 27.8% ● Go 7.3%
● Python 2.4% ● CMake 2.2%
● Shell 1.6% ● Other 2.4%

@BhosalePratim

Initializing the transcription model using the whisper.New()



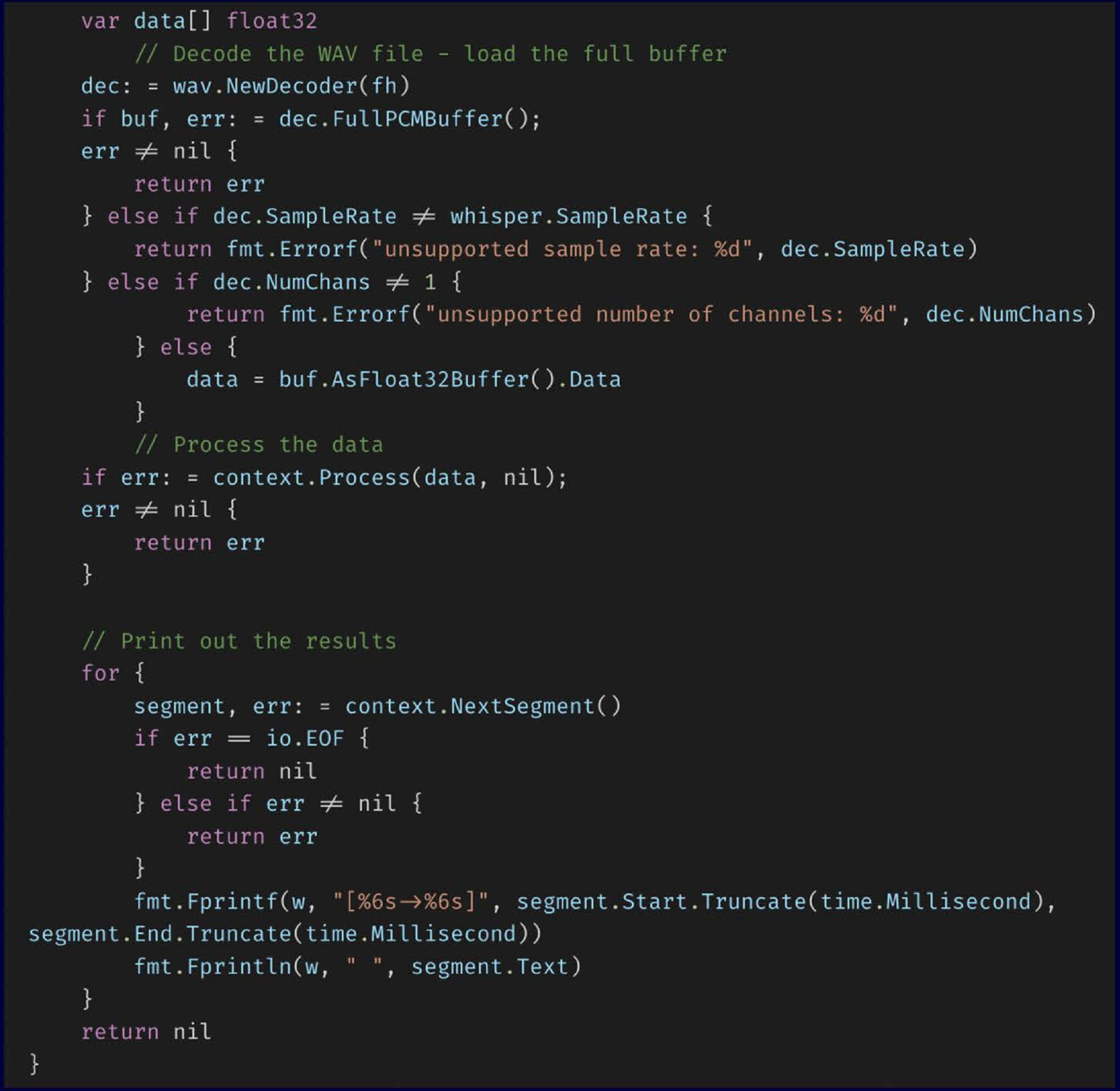
```go
func transcribe(audioFilename string, modelPath string) error {
    // load whisper model
    model, err: = whisper.New(modelPath)
    if err ≠ nil {
        return fmt.Errorf("failed to load model: %w", err)
    }
    defer model.Close()

    log.Println("Successfully loaded the model")

    // Create processing context
    context, err: = model.NewContext()
    if err ≠ nil {
        return fmt.Errorf("failed to create context: %w", err)
    }
    fh, err: = os.Open(audioFilename)
    if err ≠ nil {
        return fmt.Errorf("failed to open audio file: %w", err)
    }
    defer fh.Close()

    if err ≠ nil {
        return fmt.Errorf("failed to open audio file: %w", err)
    }
```

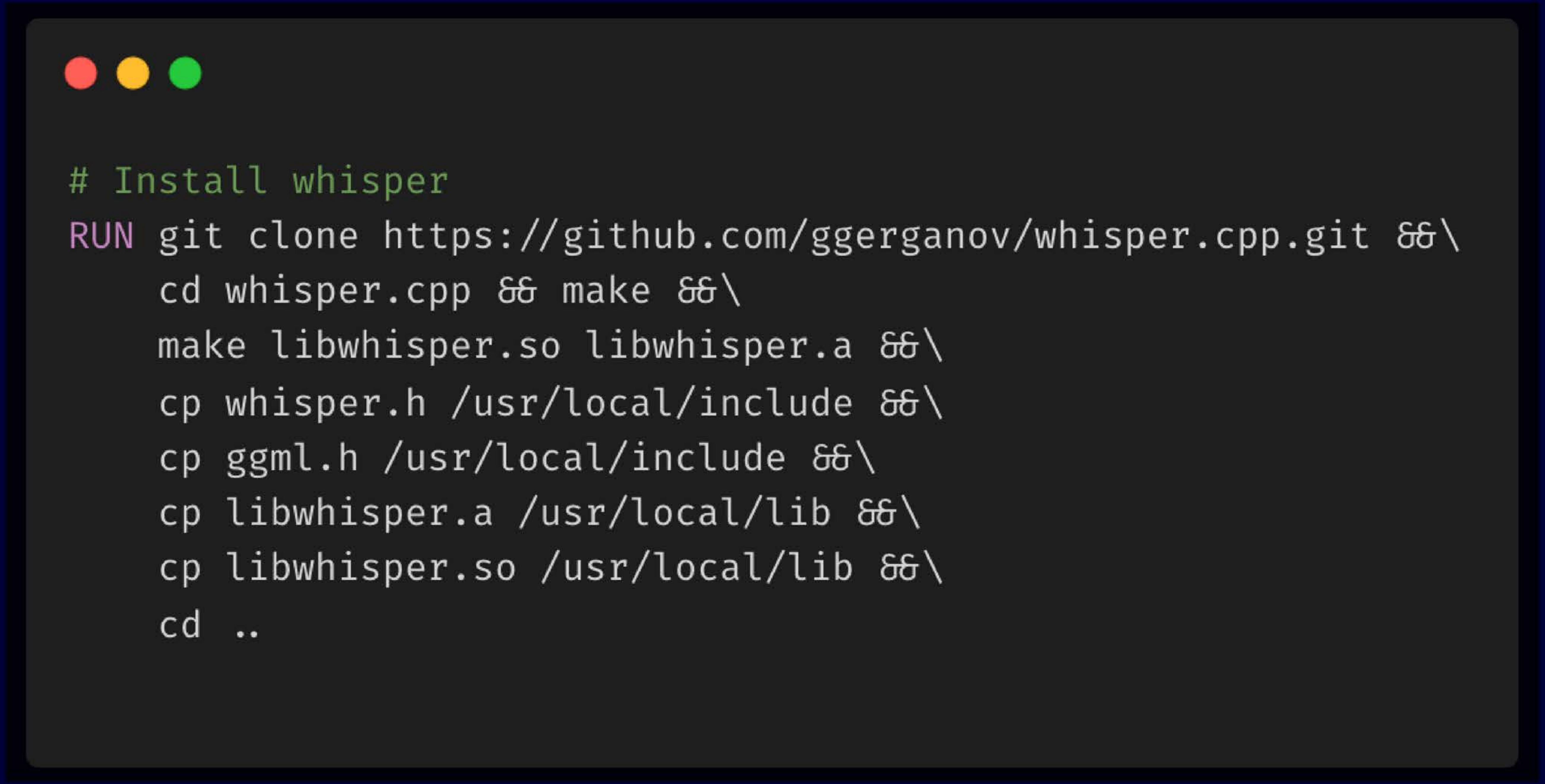
Decoding the WAV file and processing the context.



```
    var data[] float32
        // Decode the WAV file - load the full buffer
    dec: = wav.NewDecoder(fh)
    if buf, err: = dec.FullPCMBuffer();
    err ≠ nil {
        return err
    } else if dec.SampleRate ≠ whisper.SampleRate {
        return fmt.Errorf("unsupported sample rate: %d", dec.SampleRate)
    } else if dec.NumChans ≠ 1 {
            return fmt.Errorf("unsupported number of channels: %d", dec.NumChans)
        } else {
            data = buf.AsFloat32Buffer().Data
        }
        // Process the data
    if err: = context.Process(data, nil);
    err ≠ nil {
        return err
    }

    // Print out the results
    for {
        segment, err: = context.NextSegment()
        if err == io.EOF {
            return nil
        } else if err ≠ nil {
            return err
        }
        fmt.Fprintf(w, "[%6s→%6s]", segment.Start.Truncate(time.Millisecond),
segment.End.Truncate(time.Millisecond))
        fmt.Fprintln(w, " ", segment.Text)
    }
    return nil
}
```

Containerizing Whisper using Docker



```
# Install whisper
RUN git clone https://github.com/ggerganov/whisper.cpp.git &&\
    cd whisper.cpp && make &&\
    make libwhisper.so libwhisper.a &&\
    cp whisper.h /usr/local/include &&\
    cp ggml.h /usr/local/include &&\
    cp libwhisper.a /usr/local/lib &&\
    cp libwhisper.so /usr/local/lib &&\
    cd ..
```

Usecases

- Transcribing meetings
- Audio Chatbots
- Automatic Translations
- Video Subtitles

Demo

Let's take a short video from youtube and convert it into text.

Thank you!