Performance evaluation of ASR in different scenarios

Alexia

*ASR: automatic speech recognition

Background – goal of the project

Dataset 1 :
Crema_df
😀😭😡😫

Dataset 2:
All_speakers_df
🇸🇬🇬🇧🇺🇸🇰🇷🇨🇳🇫🇷

Model 1 :
Open AI's Whisper (Basic)

Model 2:
Google Speech to Text

M1D1

M2D2

M2D1

M2D2

Evaluation:

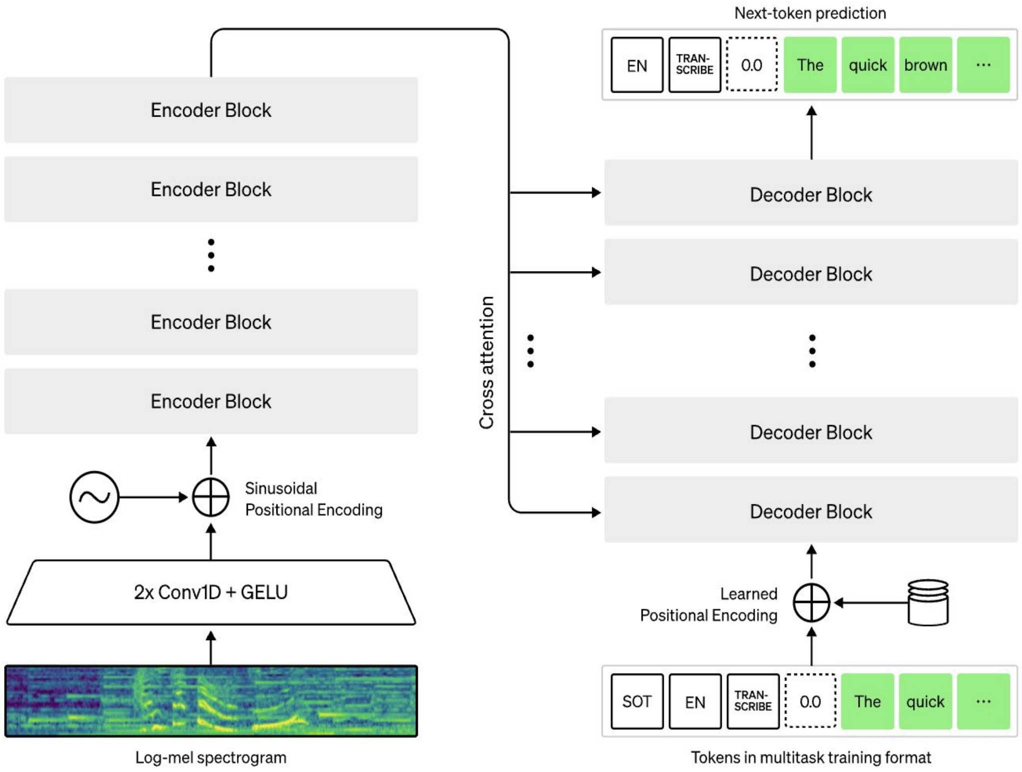
WER
BERTScore
METEOR

Background

The different philosophies in ASR:

- **Whisper** is open-source, research-focused, and designed for general robustness.
- **Google's Speech-to-Text** is a production-grade API optimized for real-time accuracy and scalability.

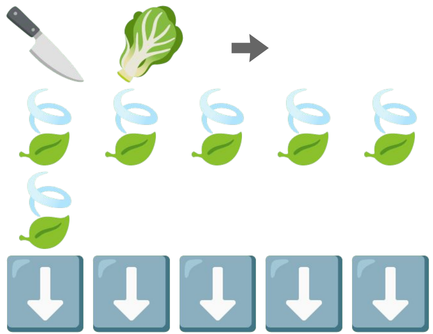
Background – Open AI's Whisper



2022; 680kh

Transformer-based encoder-decoder model

30-second chunks → log-Mel spectrogram



Background – Open AI’s Whisper

Multitask:

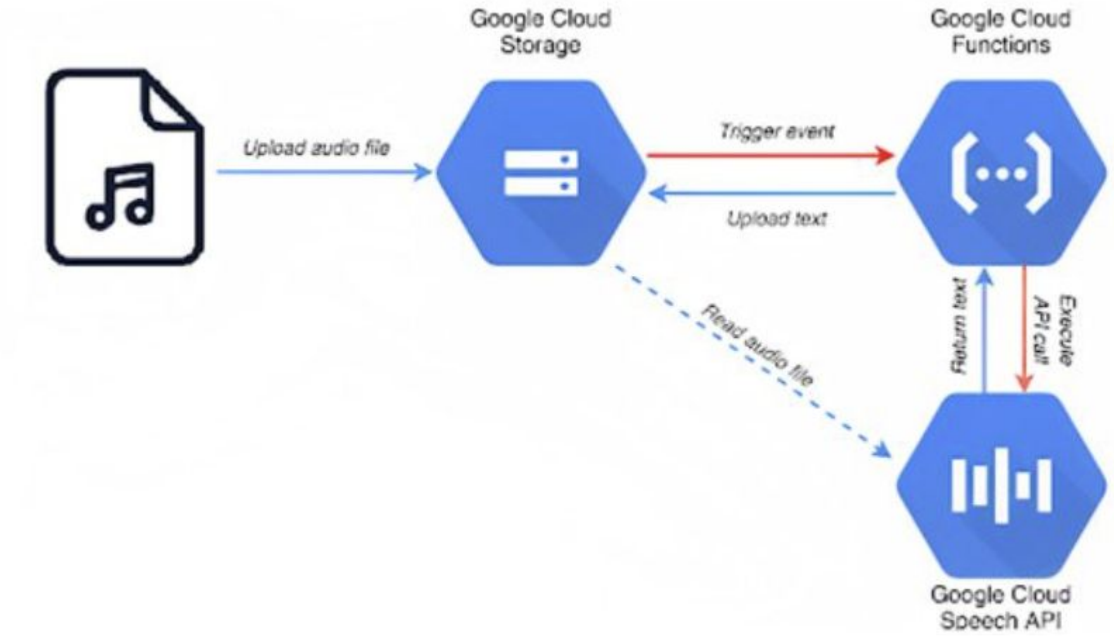
- Detect language,
- Generate timestamps,
- Translate speech into English from other languages.

Open-source :

- free
- can run it locally on our own machines
- full control over privacy and fine-tuning.

Slow & not optimized for streaming.

Background – Google Speech to Text



cloud-based API service

Fast, scalable

Background

Conformer Transducer:

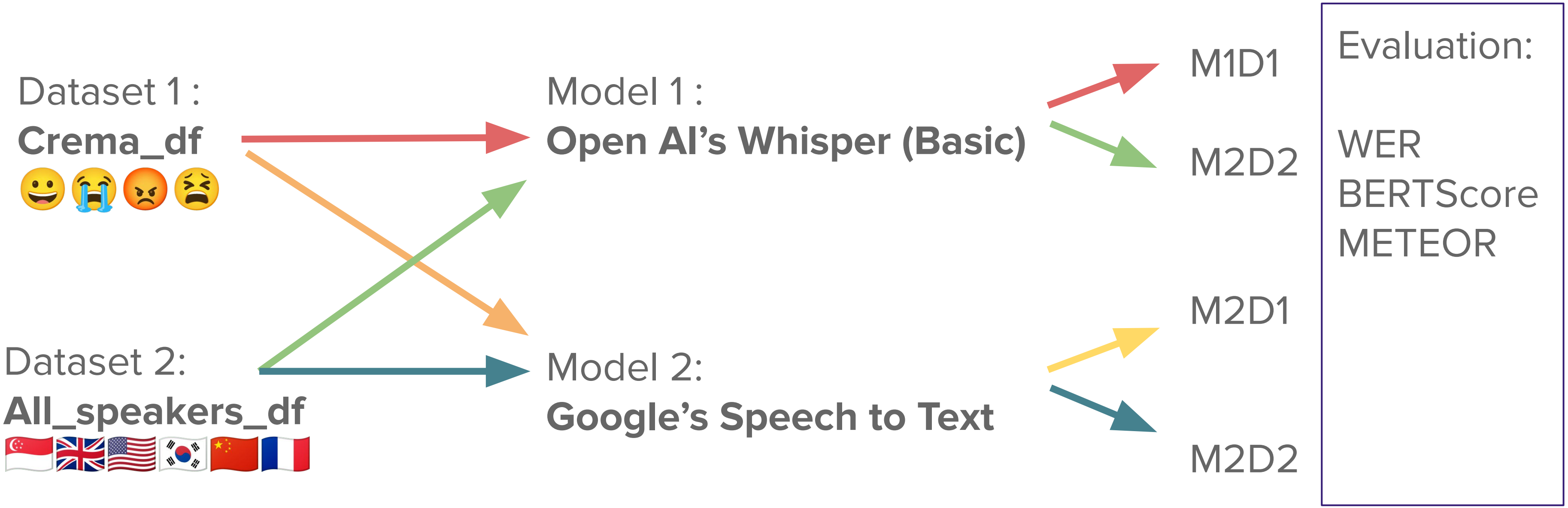
- **Convolutional layers**— to capture local audio features
- **Transformer layers**— to understand the broader context.
- Optimized for **real-time transcription**

Supports over 100 languages

Offers domain-specific customization with things like phrase hints.

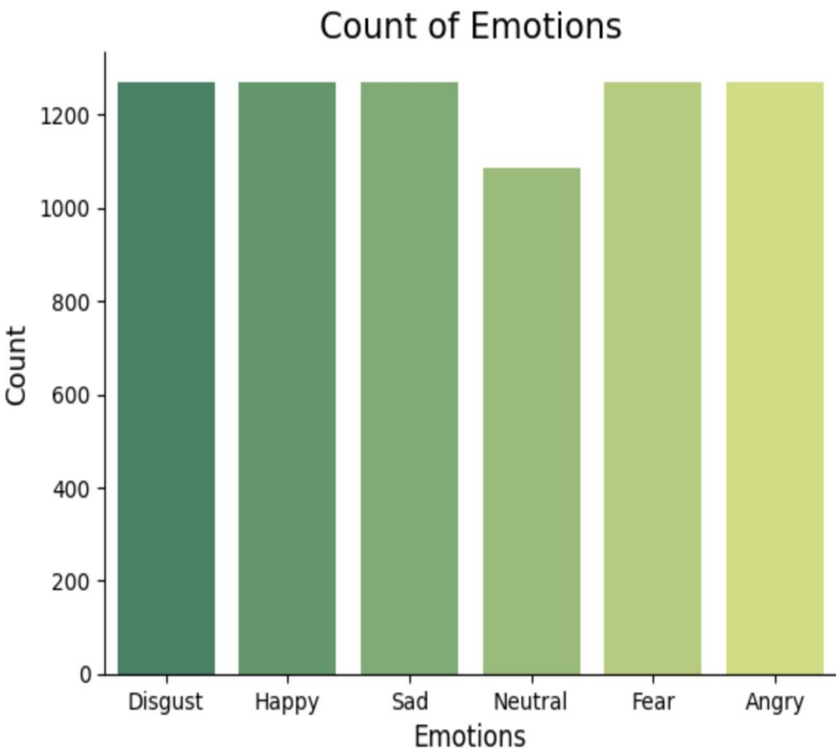
Paid service & cloud-based — privacy trade-offs & usage limits.

Background – goal of the project



Dataset 1 : **Crema_df** 😀😭😡😫

D1: Data exploration



From 91 actors

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7442 entries, 0 to 7441
Data columns (total 2 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Emotions   7442 non-null   object
 1   Path       7442 non-null   object
dtypes: object(2)
memory usage: 116.4+ KB
```

	Emotions	Path
0	Disgust	/kaggle/input/cremad/AudioWAV/1028_TSI_DIS_XX.wav
1	Happy	/kaggle/input/cremad/AudioWAV/1075_IEO_HAP_LO.wav
2	Happy	/kaggle/input/cremad/AudioWAV/1084_ITS_HAP_XX.wav

D1: Data exploration

```
# Initialize new column
df['Transcript_Human'] = None

# Define pattern-transcript mappings
pattern_transcript = {
    '_DFA_': "Don't forget the jacket.",
    '_IEO_': "It’s 11 o’clock",
    '_IOM_':"I'm on my way to the meeting.",
    '_ITH_':"I think I have a doctor's point.",
    '_ITS_':"I think I have seen this before.",
    '_IWL_':"I would like a new alarm clock.",
    '_IWW_':"I wonder what this is about.",
    '_MTI_':"Maybe tomorrow it will be cold.",
    '_TAI_':"The airplane is almost full.",
    '_TIE_':"That is exactly what happened.",
    '_TSI_':"The surface is slick.",
    '_WSI_':"We'll stop in a couple of minutes.",
}

# Apply mappings
for pattern, transcript in pattern_transcript.items():
```

Angry

Happy

Disgusting

Neutral

Fear

Sad

D2: Data exploration

Dataset 2:

All_speakers_df

🇸🇬🇬🇧🇺🇸🇰🇷🇨🇳🇫🇷

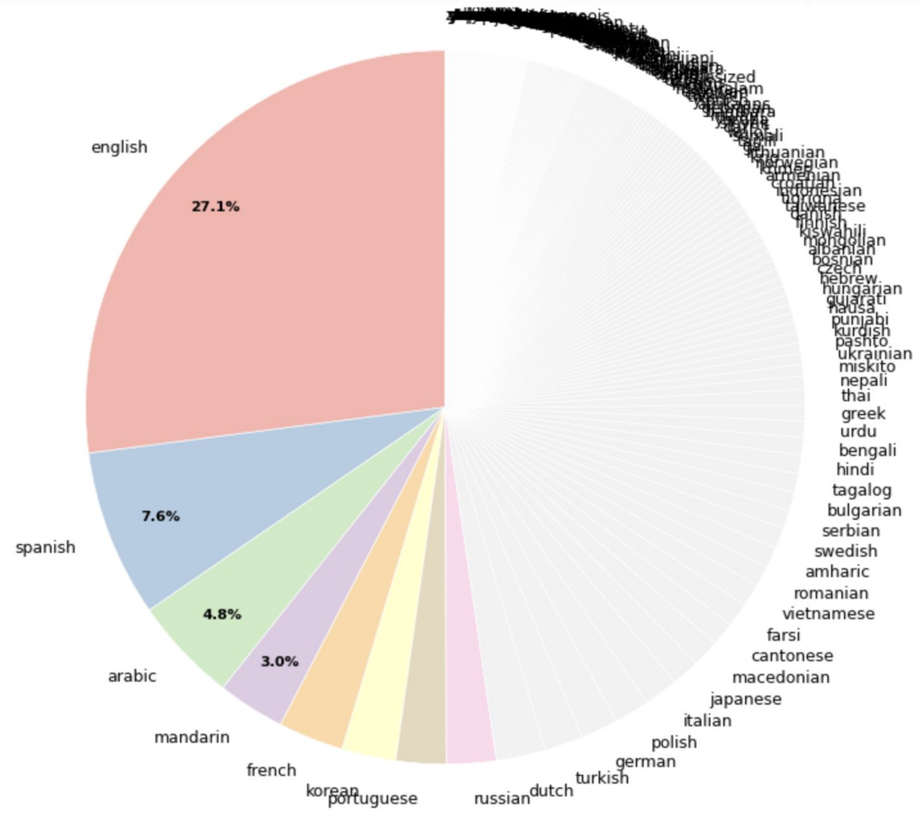
```
<class 'pandas.core.frame.DataFrame'>
Index: 2140 entries, 32 to 2171
Data columns (total 12 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   age              2140 non-null   float64
 1   age_onset        2140 non-null   float64
 2   birthplace       2136 non-null   object 
 3   filename         2140 non-null   object 
 4   native_language  2140 non-null   object 
 5   sex              2140 non-null   object 
 6   speakerid        2140 non-null   int64  
 7   country          2135 non-null   object 
 8   file_missing?    2140 non-null   bool   
 9   Unnamed: 9       0 non-null      float64
 10  Unnamed: 10      0 non-null      float64
 11  Unnamed: 11      1 non-null      object 
dtypes: bool(1), float64(4), int64(1), object(6)
memory usage: 202.7+ KB
```

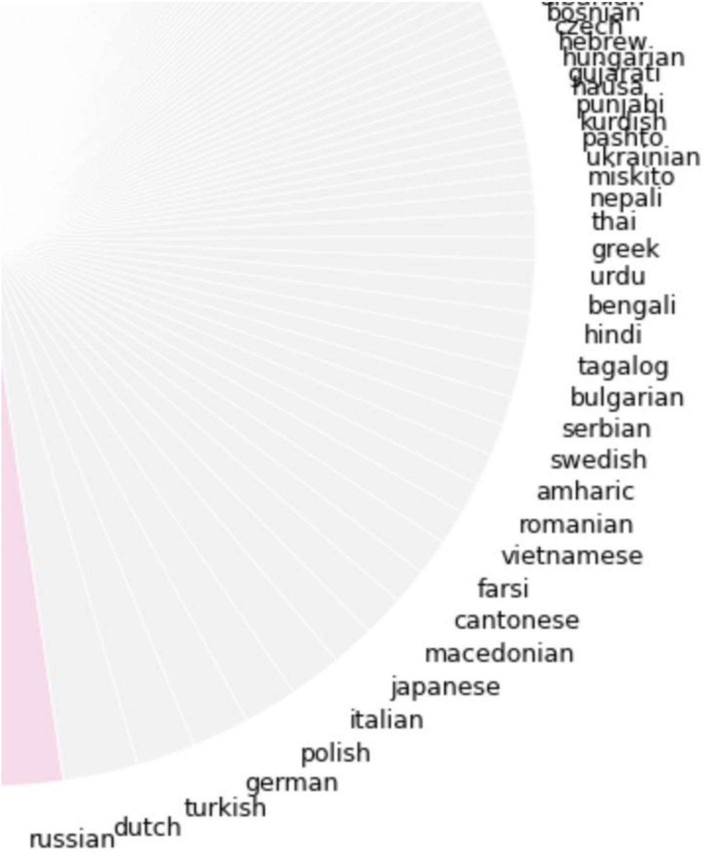
	age	age_onset	birthplace	filename	native_language	sex	speakerid	country	file_missing?	Unnamed: 9	Unnamed: 10	Unnamed: 11	Path
32	27.0	9.0	virginia, south africa	afrikaans1	afrikaans	female	1	south africa	False	NaN	NaN	NaN	/kaggle/input/speech-accent-archive/recordings...
33	40.0	5.0	pretoria, south africa	afrikaans2	afrikaans	male	2	south africa	False	NaN	NaN	NaN	/kaggle/input/speech-accent-archive/recordings...
34	43.0	4.0	pretoria, transvaal, south africa	afrikaans3	afrikaans	male	418	south africa	False	NaN	NaN	NaN	/kaggle/input/speech-accent-archive/recordings...

D2: Data exploration

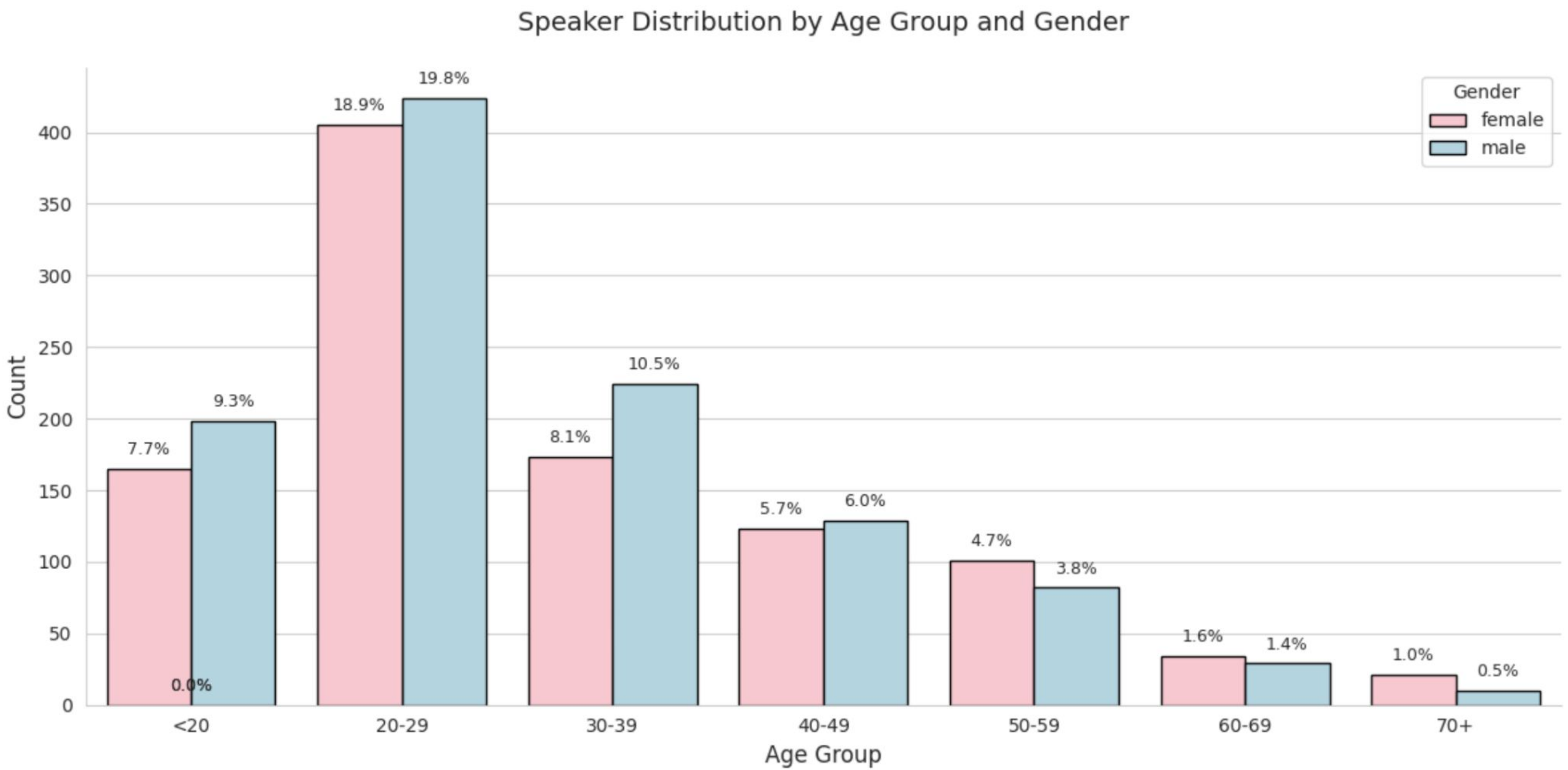
“Please call Stella. Ask her to bring these things with her from the store: Six 🥄 of fresh snow peas, ⑤ thick slabs of 🔵 🧀, and maybe a snack for her brother Bob. We also need a small plastic 🐍 and a big toy 🐸 for the kids. She can scoop these things into three 🔴 bags, and we will go meet her Wednesday at the train station.”

D2: Data exploration – native language distribution

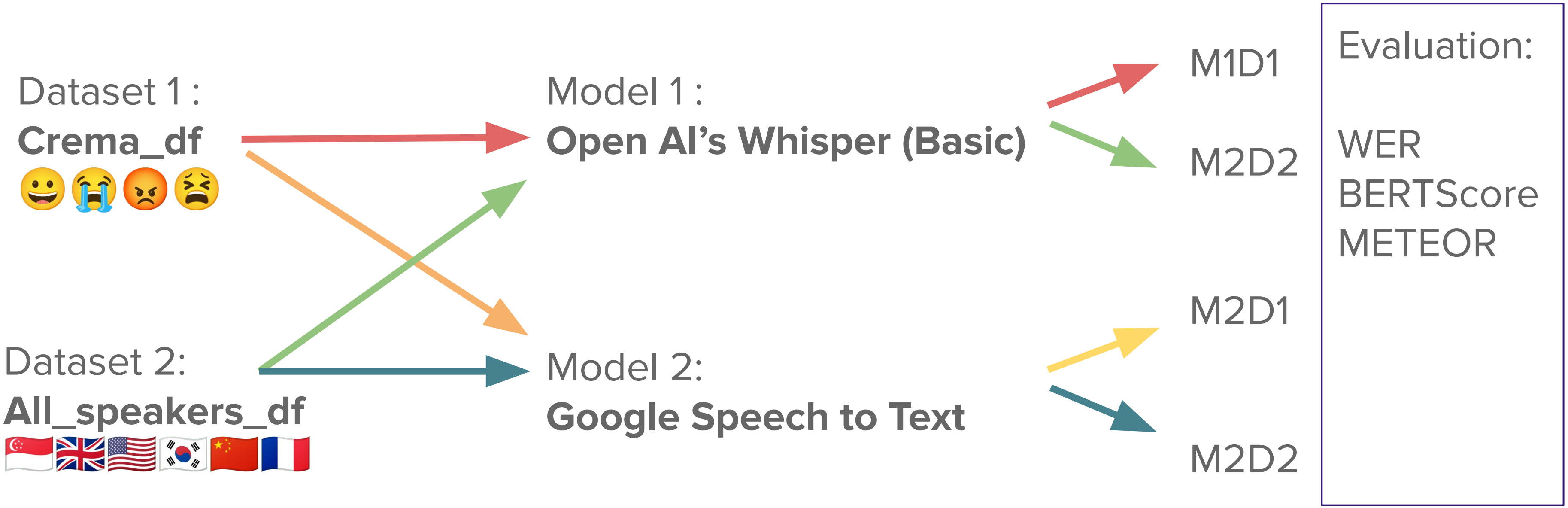




D2: Data exploration



Background – goal of the project



M1D1

```python
from transformers import pipeline
import torchaudio
import torch
from tqdm import tqdm

# load ASR model (Whisper)
asr = pipeline("automatic-speech-recognition", model="openai/whisper-base", device=0 if torch.cuda.is_available() else -1)

# Create function to transcribe with error handling
def safe_transcribe(path):
    try:
        return asr(path)["text"]
    except Exception as e:
        print(f"Error on {path}: {e}")
        return ""

# Apply with progress bar
tqdm.pandas()
crema_df['Transcript'] = crema_df['Path'].progress_apply(safe_transcribe)
```

M1D1

	Emotions	Path	Transcript	Transcript_Human
0	Disgust	/kaggle/input/cremad/AudioWAV/1028_TSI_DIS_XX.wav	This surface is slick.	The surface is slick.
1	Happy	/kaggle/input/cremad/AudioWAV/1075_IEO_HAP_LO.wav	It's 11 o'clock.	It’s 11 o’clock
2	Happy	/kaggle/input/cremad/AudioWAV/1084_ITS_HAP_XX.wav	I think I've seen this before.	I think I have seen this before.
3	Disgust	/kaggle/input/cremad/AudioWAV/1067_IWW_DIS_XX.wav	I wonder what this is about.	I wonder what this is about.
4	Disgust	/kaggle/input/cremad/AudioWAV/1066_TIE_DIS_XX.wav	That is exactly what happened.	That is exactly what happened.

M1D2

Path	Transcript_Corr	M1_transcript
/kaggle/input/speech-accent-archive/recordings...	Please call Stella. Ask her to bring these thi...	Please call Stella, ask her to bring these th...
/kaggle/input/speech-accent-archive/recordings...	Please call Stella. Ask her to bring these thi...	Please call Stella, ask her to bring these th...
/kaggle/input/speech-accent-archive/recordings...	Please call Stella. Ask her to bring these thi...	Please call Stella, ask her to bring these th...

M2D1

Alexia's Project

```
[ ] from google.colab import files
    uploaded = files.upload()
```

Choose files No file chosen Upload widget is only available when the c
to enable.
Saving molten-complex-457317-d9-dab85b056fab.json to molten-co

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibl

```
[ ] import os

    audio_dir = "/content/drive/MyDrive/AudioWAV/"
    if os.path.exists(audio_dir):
        print("Files in AudioWAV folder:")
        print(os.listdir(audio_dir)[:10])  # Print first 10 files
    else:
        print("AudioWAV directory not found!")
```

Files in AudioWAV folder:
['1079_WSI_DIS_XX.wav', '1079_TIE_FEA_XX.wav', '1079_TIE_ANG_X

```
from google.cloud import speech
from tqdm import tqdm

client = speech.SpeechClient()

def transcribe_audio(file_path):
    flac_path = convert_wav_to_flac(file_path)
    with open(flac_path, "rb") as audio_file:
        content = audio_file.read()

    audio = speech.RecognitionAudio(content=content)
    config = speech.RecognitionConfig(
        encoding=speech.RecognitionConfig.AudioEncoding.FLAC,
        sample_rate_hertz=16000,  # adjust if different
        language_code="en-US"
    )

    try:
        response = client.recognize(config=config, audio=audio)
        return " ".join([result.alternatives[0].transcript for result in response.results])
    except Exception as e:
        print(f"Error for {file_path}: {e}")
        return ""
```

M2D1


```python
# Extract speaker ID from the path (corrected regex)
crema_df['Speaker'] = crema_df['FileName'].str.extract(r'^(\d+)_')
crema_df_filtered = crema_df[crema_df['Speaker'].isin(['1001', '1002', '1003', '1004', '1005'])]
```

Total duration of the audio = 12 x 6 x 5 x 2.5 = 900 seconds
(12 Reading passage *6 emotions * 5 actors * average audio length)

	FileName	Path	Transcript	Speaker	Transcript_Human
7000	1001_DFA_ANG_XX.wav	/content/drive/MyDrive/AudioWAV/1001_DFA_ANG_X...	don't forget a jacket	1001	Don't forget the jacket.
7001	1001_DFA_DIS_XX.wav	/content/drive/MyDrive/AudioWAV/1001_DFA_DIS_X...	don't forget a jacket	1001	Don't forget the jacket.
7002	1001_DFA_HAP_XX.wav	/content/drive/MyDrive/AudioWAV/1001_DFA_HAP_X...	don't forget a jacket	1001	Don't forget the jacket.

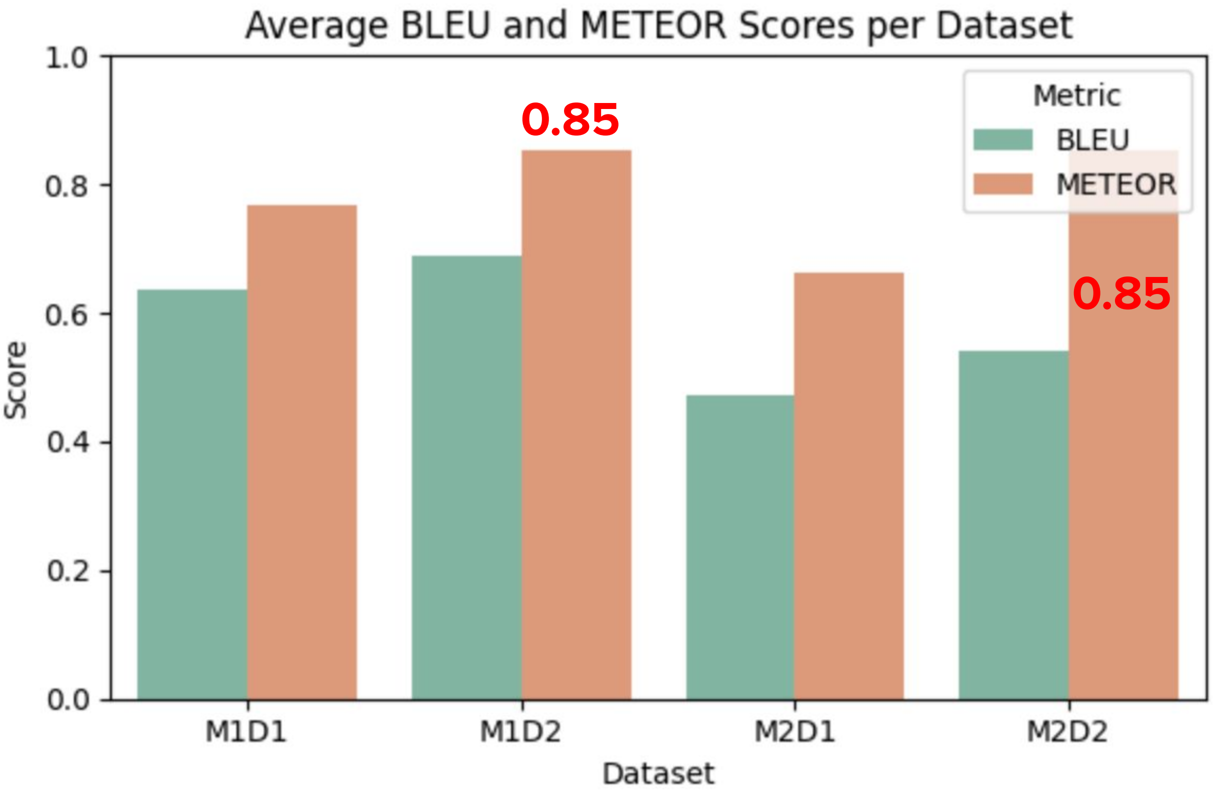
M2D2

Asp_filtered = asp_df.head(40)

Total duration of the audio = 30 x 40=1200 seconds

Path	Transcript_Corr	Transcript_GG
/content/drive/MyDrive/AudioWAV2/wav/afrikaans...	Please call Stella. Ask her to bring these thi...	please call Stella asked her to bring these th...
/content/drive/MyDrive/AudioWAV2/wav/afrikaans...	Please call Stella. Ask her to bring these thi...	please call Stella asked her to bring these th...
/content/drive/MyDrive/AudioWAV2/wav/afrikaans...	Please call Stella. Ask her to bring these thi...	please call Stella asked her to bring these th...

Evaluation



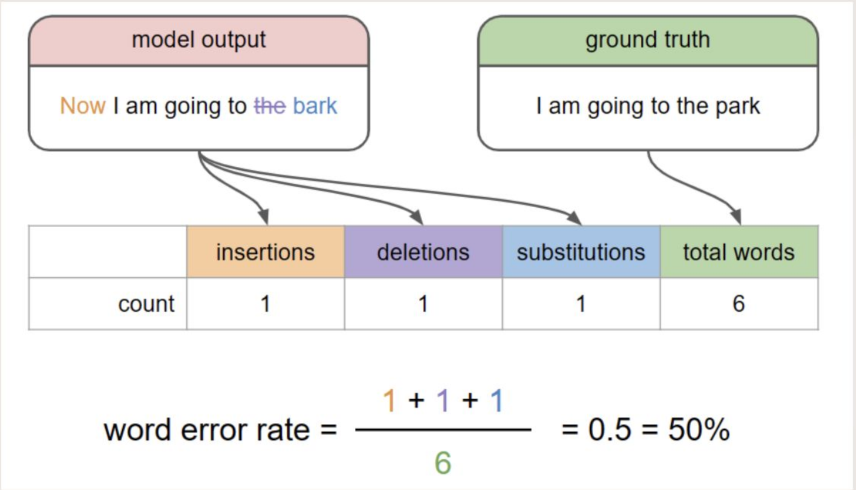
METEOR

- **Metric for Evaluation of Translation with Explicit ORdering**
- evaluates based on **[Precision + Recall], Synonym matching, Stemming**, and **Word order.**

Evaluation

Word Error Rate (WER)

- most widely used metric for evaluating speech-to-text models.
- percentage of errors (insertions, deletions, and substitutions) made by the model in its transcript compared to a human-generated ground truth transcript.
- A lower WER indicates better accuracy.



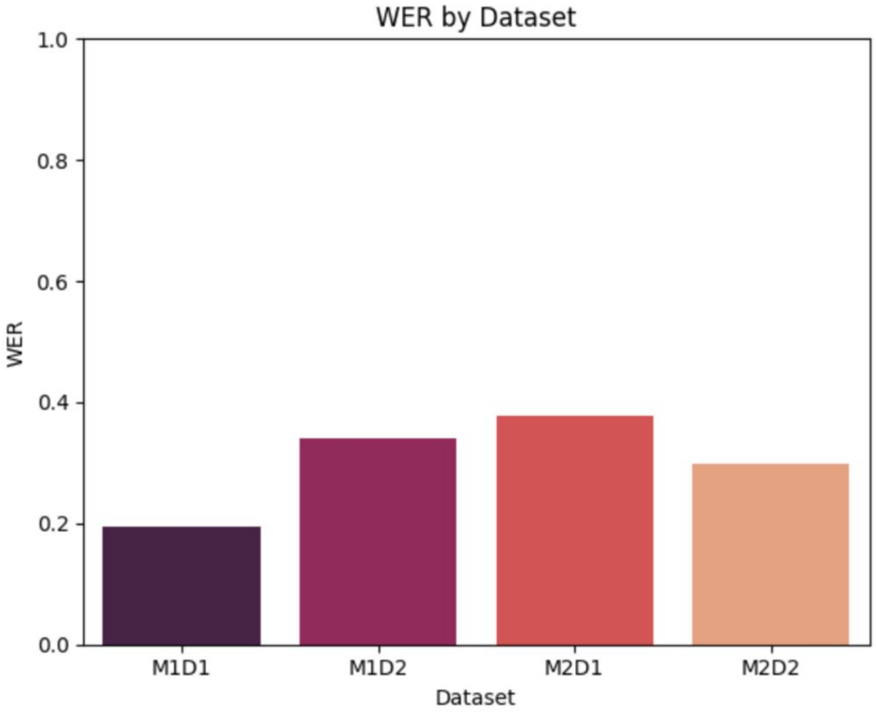
Calculation of WER for an example model output given the ground truth transcription

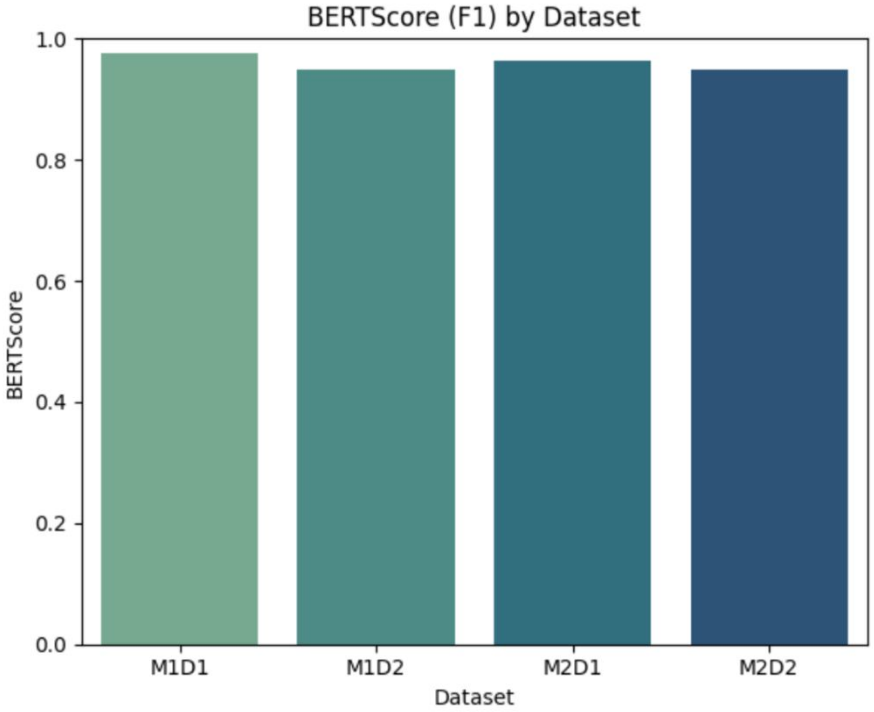
BertScore:

A metric that uses pre-trained language model embeddings to calculate the cosine similarity between the generated text and the reference text

Evaluation

	Dataset	WER	BERTScore
0	M1D1	0.193777	0.976976
1	M1D2	0.341617	0.948523
2	M2D1	0.378032	0.963611
3	M2D2	0.298188	0.949644





Ethical Evaluation

70-80% accuracy != scientific excellence (Google's principles)

Offers notable social advantages.

Provides valuable insights into appropriate deployment contexts and limitations

Limitations & Further Improvements

- Dataset Diversity (recordings of conversations among multiple individuals, audio from songs, etc…)
- Implement supplementary technical metrics alongside human evaluations
- Model Diversity
- Appropriate safeguard — providing opt-out options

Model	CNN Usage	Main Architecture
Wav2Vec 2.0	Feature extraction	CNN + Transformer
DeepSpeech 2	Initial layers	CNN + Bi-LSTM
Whisper	Encoder front-end	CNN + Transformer
Listen, Attend, Spell (LAS)	Optional preprocessing	CNN + RNN + Attention


THANK YOU