Automated Speech Recognition and Intent Classification

MIND-14 Dataset

Kaira Milani Fitria

08 November 2024

Meet the Team

Kaira Milani Fitria

Team Member

Kevin Sutanto

Team Member

Introduction

History

1. Cikal Bakal (1950 - 1960)
Audrey - Bell Labs, Shoebox - IBM
2. Era Pengembangan Awal (1970-1980)
DARPA (Defense Advanced Research Projects Agency) - US
Harpy - Carneige Mellon
3. Kemajuan Signifikan (1980-1990)
Tangora - IBM, Julie - World of Wonder
4. Era Komersial Awal (1990-2000)
Dragon Dictate - Dragon, Dragon NaturallySpeaking
5. Era Modern (2000-sekarang)
Microsoft, Google, Siri - Apple, Alexa - Amazon, dll

Background

1. Peningkatan Interaksi Digital
2. Aksesbilitas
3. Digitalisasi Informasi

ASR Implementation

1. Asisten Virtual dan Pengenalan Suara
2. Layanan Pelanggan Otomatis
3. Transkripsi Otomatis
4. Pengenalan Ucapan dalam Kendaraan
5. Aksesibilitas untuk Penyandang Disabilitas
6. Sistem Penerjemahan Bahasa Otomatis
7. Industri Kesehatan
8. Pencarian Suara (Voice Search)

Data Preparation

MINDS-14 is a dataset designed for the intent detection task with spoken data. It encompasses 14 distinct intents extracted from a commercial system in the e-banking domain.

```
!huggingface-cli download PolyAI/minds14 --repo-type dataset --revision refs/convert/parquet --local-dir . --local-dir-use-symlinks False  --include 'en-US/*'
minds_enUS = load_dataset('./en-US', split="train")
```

```
Dataset({
    features: ['path', 'audio', 'transcription', 'english_transcription', 'intent_class', 'lang_id'],
    num_rows: 563
})
```

```
{
    "path": ".../en-US~JOINT_ACCOUNT/602ba55abb1e6d0fbce92065.wav",
    "audio": {
        "path": "602ba55abb1e6d0fbce92065.wav",
        "array": array(
                [ 0., 0.00024414, -0.00024414,..., -0.00024414,0.,  0.]
        ),
        "sampling_rate": 8000,
    },
    "transcription": "I would like to set up a joint account with my partner",
    "english_transcription": "I would like to set up a joint account with my partner",
    "intent_class": 11,
    "lang_id": 4,
}
```

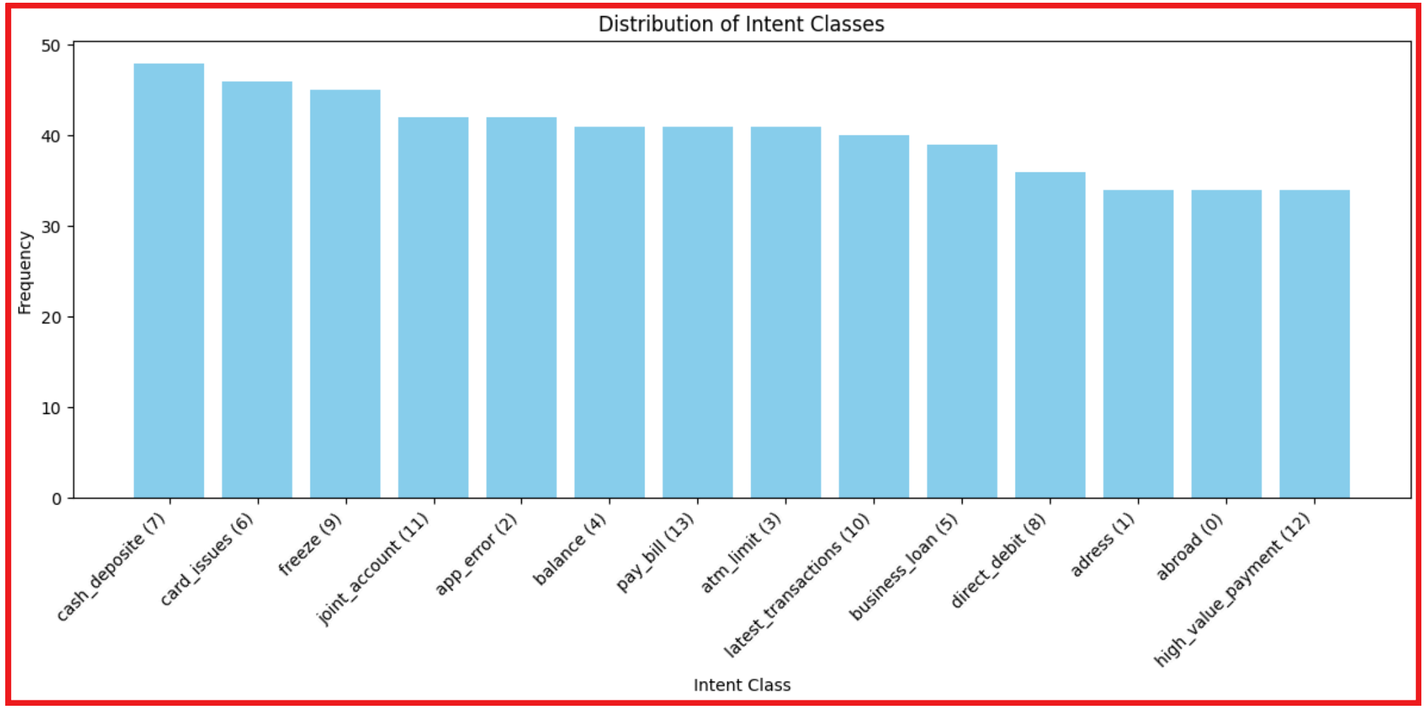
- **path** : full file path to the audio file
- **audio** : dictionary sub-elements audio data
    - ***path***: path to the audio file.
    - ***array***: array of raw audio samples represent waveform.
    - ***sampling_rate***: audio was sampled at 8000 samples per second
- **transcription** : original transcription of audio
- **english_transcription** : english translation of the original transcription
- **intent_class** : numeric label representing the intent behind the audio
- **lang_id** : numeric identifier for the language of the audio

Data Preparation

```
intent_classes = [
    {'index': 0, 'name': 'abroad'},
    {'index': 1, 'name': 'adress'},
    {'index': 2, 'name': 'app_error'},
    {'index': 3, 'name': 'atm_limit'},
    {'index': 4, 'name': 'balance'},
    {'index': 5, 'name': 'business_loan'},
    {'index': 6, 'name': 'card_issues'},
    {'index': 7, 'name': 'cash_deposite'},
    {'index': 8, 'name': 'direct_debit'},
    {'index': 9, 'name': 'freeze'},
    {'index': 10, 'name': 'latest_transactions'},
    {'index': 11, 'name': 'joint_account'},
    {'index': 12, 'name': 'high_value_payment'},
    {'index': 13, 'name': 'pay_bill'}
]
```

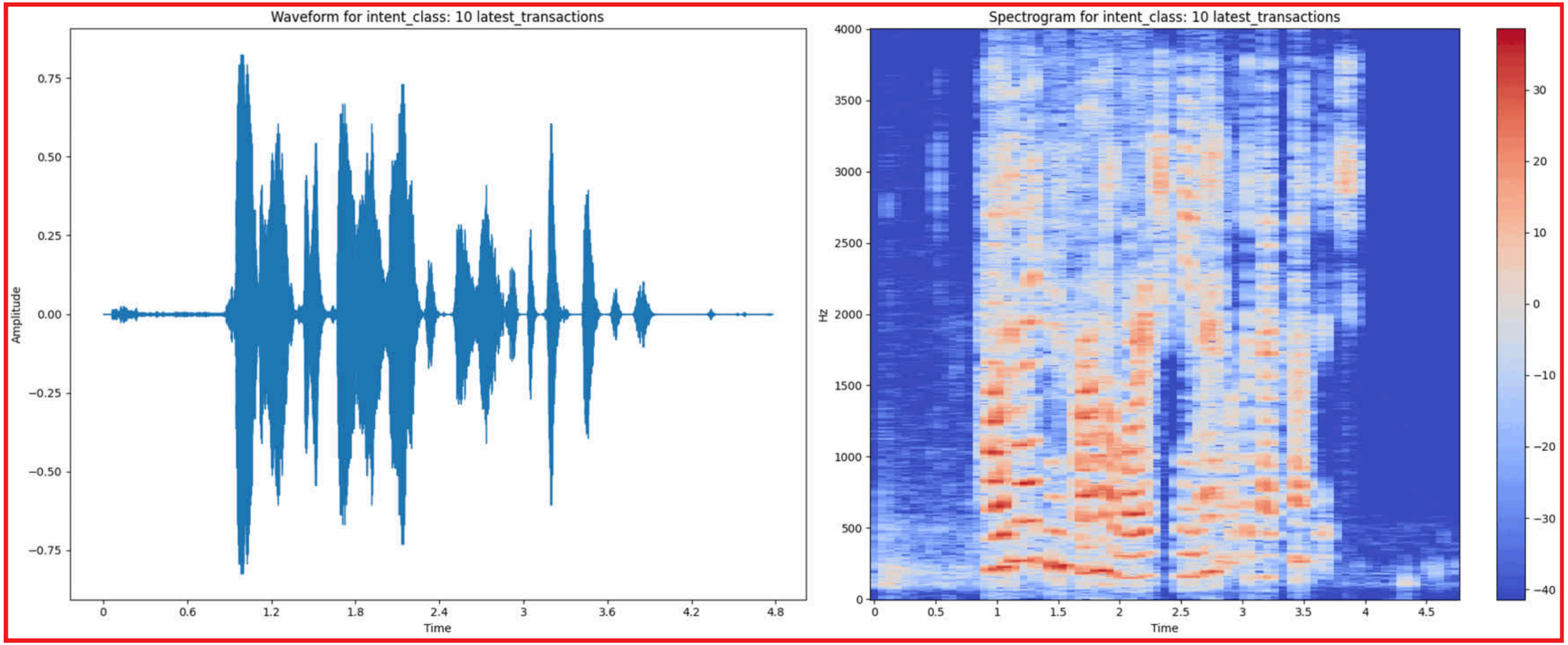
MINDS-14 is training and evaluation resource for intent detection task with spoken data. It covers 14 intents extracted from a commercial system in the e-banking domain

The intent_class label provides supervised data for training models that predict the speaker's intent.



The frequencies for each intent class appear similar, ranging between 40 to 50 occurrences per class.
Intent distribution in this dataset is balanced and suitable for training an intent classification model.

Data Preparation



The audio file is also loaded into memory as arrays for easier handling and faster processing. this also perform transformations, feature extractions (like MFCCs or spectrograms), and model inputs without repeatedly decoding .wav files, which can be slow.

Data Preparation

Check Null Values

```
{'path': 0, 'audio': 0, 'transcription': 0, 'english_
transcription': 0, 'intent_class': 0, 'lang_id': 0}
```

Remove Special Character

```
def remove_special_characters(batch):
    batch["transcription"] = re.sub([\,\?\.\!\-\;\:\"], '',
    batch["transcription"]).lower() + " "
    batch["transcription"] = batch['transcription'].rstrip()
    return batch
```

Train Test Split

```
DatasetDict({
    train: Dataset({
        features: ['path', 'audio', 'transcription', 'english_transcription',
        'intent_class', 'lang_id'],
        num_rows: 450
    })
    test: Dataset({
        features: ['path', 'audio', 'transcription', 'english_transcription',
        'intent_class', 'lang_id'],
        num_rows: 113
    })
})
```

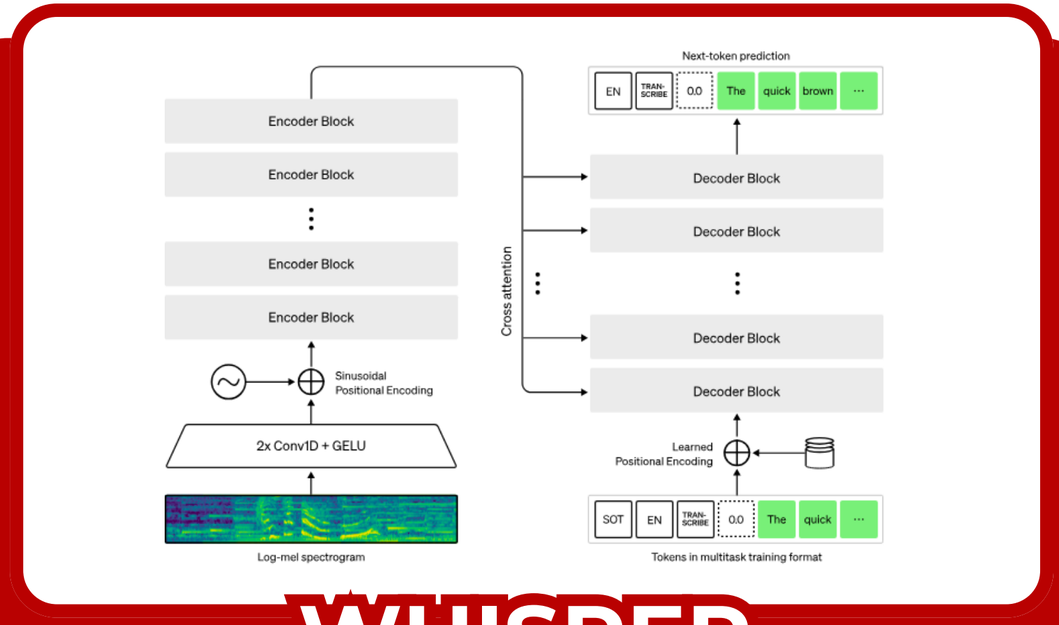
Extract Words & Build Vocabulary

```
def extract_all_chars(batch):
  all_text = " ".join(batch["transcription"])
  vocab = list(set(all_text))
  return {"vocab": [vocab], "all_text": [all_text]}

vocabs = dataset.map(
    extract_all_chars, batched=True,
    batch_size=1, keep_in_memory=True)

vocab_list = list(set(vocabs["train"]["vocab"][0]) | set(vocabs["test"]["vocab"][0]))
vocab_dict = {v: k for k, v in enumerate(vocab_list)}
print(vocab_dict)
print(len(vocab_dict))
```

```
{'g': 0, 's': 1, 't': 2, 'y': 3, 'd': 4, 'n': 5, 'e': 6, 'h': 7, 'k': 8, 'v': 9, 'c': 10, 'u': 11,
'w': 12, 'm': 13, 'a': 14, 'r': 15, 'l': 16, 'i': 17, 'o': 18, "'": 19, ' ': 20, 'b': 21, 'p': 22} 25
```

- Vocabulary dictionary (vocab_dict) can be used to **encode transcriptions** into **numerical sequences** for training model
- Each **character** has a unique **integer** representation
- example : the word "**STOP**" would be converted to **[1, 2, 18, 22]** using this dictionary

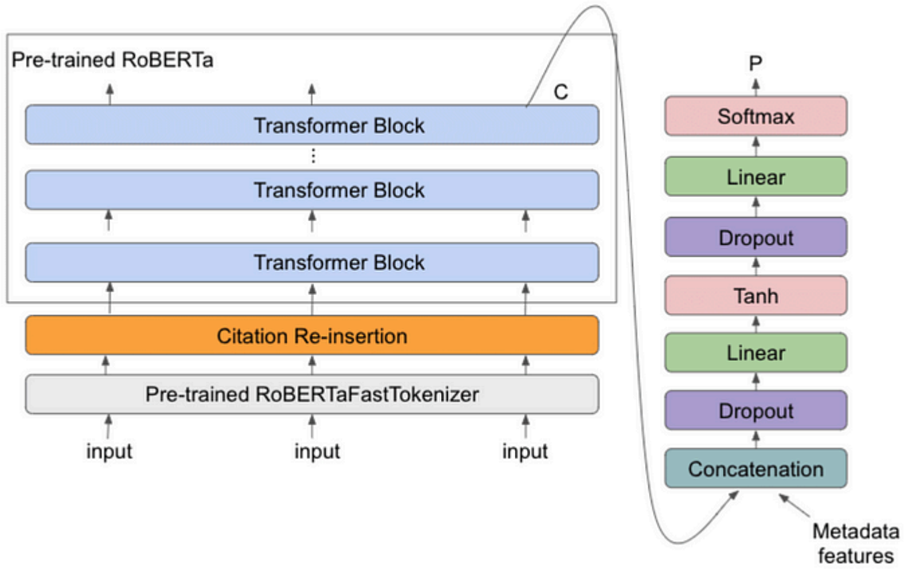
Model Development



WHISPER

Encoder-Decoder Transformer + Audio Feature Extraction + Attention Mechanism

- Transformer encoder processes audio features and the decoder generates text based on these features.
- This model allows it to capture the context of sentences, which is valuable in conversational ASR tasks.



RoBERTa

Transformer Encoder + Attention Mechanism + Output Layer Classifier

- Based on a transformer encoder, uses attention to understand word context within a sentence
- A dense output layer is added on top, mapping the encoded text to intent labels and producing the most likely intent as the final output.

ASR-Pre-Processing

Model and Processor Initialization

```
whisper = 'openai/whisper-tiny'
tokenizer = WhisperTokenizer.from_pretrained(whisper, language="english")
feature_extractor = WhisperFeatureExtractor.from_pretrained(whisper)
processor = WhisperProcessor(
    feature_extractor=feature_extractor, tokenizer=tokenizer)
model = WhisperForConditionalGeneration.from_pretrained(whisper)
```

Tokenizer: Converts text into tokens
Feature Extractor: Processes raw audio into a suitable form for model
Processor: Combines the tokenizer and feature extractor into one pipeline for easier use.

Data Encoding

```
encoded_input = processor(
    audio_array, sampling_rate=audio_sr, return_tensors='pt').input_features
encoded_label = processor(
    text=audio_text, return_tensors='pt').input_ids
```

```
array:  [ 2.26674281e-04  3.08584742e-04  2.64169619e-04 ...  1.27390027e-04
 -2.77563231e-07 -3.74693336e-05]
text:  i was calling to ask about a business loan
##################################################
feature:  tensor([[[-0.7668, -0.7293, -0.4278,  ..., -0.8408, -0.8408, -0.8408],
         [-0.8408, -0.8408, -0.3879,  ..., -0.8408, -0.8408, -0.8408],
         [-0.7263, -0.8408, -0.4029,  ..., -0.8408, -0.8408, -0.8408],
         ...,
         [-0.8408, -0.8408, -0.8408,  ..., -0.8408, -0.8408, -0.8408],
         [-0.8408, -0.8408, -0.8408,  ..., -0.8408, -0.8408, -0.8408],
         [-0.8408, -0.8408, -0.8408,  ..., -0.8408, -0.8408, -0.8408]]])
label:  tensor([[50258, 50259, 50363,    72,   390,  5141,   220,  1353,  1029,   466,
           257,  1606, 10529, 50257]])
```

Data Encoding: Prepare audio and text data for input into the model.
The audio is passed through the processor to extract features.
The transcription text is tokenized to create input IDs (tokens) that the model can understand as labels.

Feature Representation as **input**
feature is 3D array (tensor) with dimensions: **[batch_size, num_frames, num_features]**. Each element inside the tensor (like -0.7668, -0.7293, etc.) represents the amplitude or energy of specific frequency bands for each frame in the audio

Label Representation :
tokenized version of the transcription

ASR-Modeling

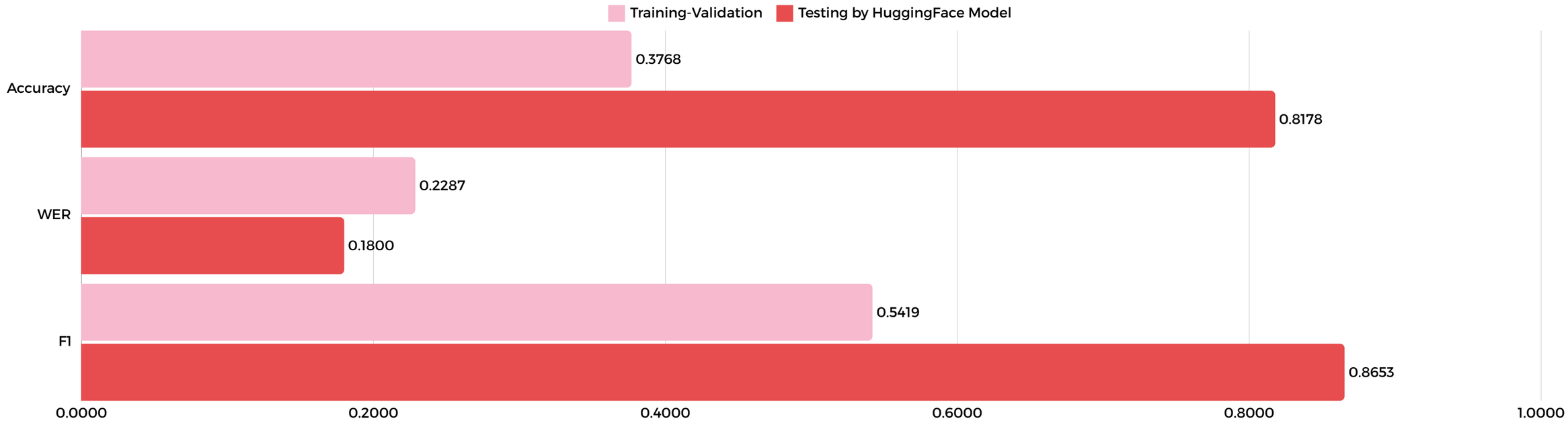
```
training_args = Seq2SeqTrainingArguments(
    output_dir="./whisper-tiny-minds14-enUS",
    per_device_train_batch_size=16,
    learning_rate=3e-5, max_steps=4000,
    fp16=True, evaluation_strategy="steps",
    save_steps=400, eval_steps=400,
    metric_for_best_model="wer")
```

openai/whisper-tiny

Automatic Speech Recognition • Updated Feb 29 • 424k • ♡ 251

kairaamilanii/whisper-mind14-enUS

Automatic Speech Recognition • Updated 5 days ago • 78



Intent Classifier-Modeling

FacebookAI/roberta-base

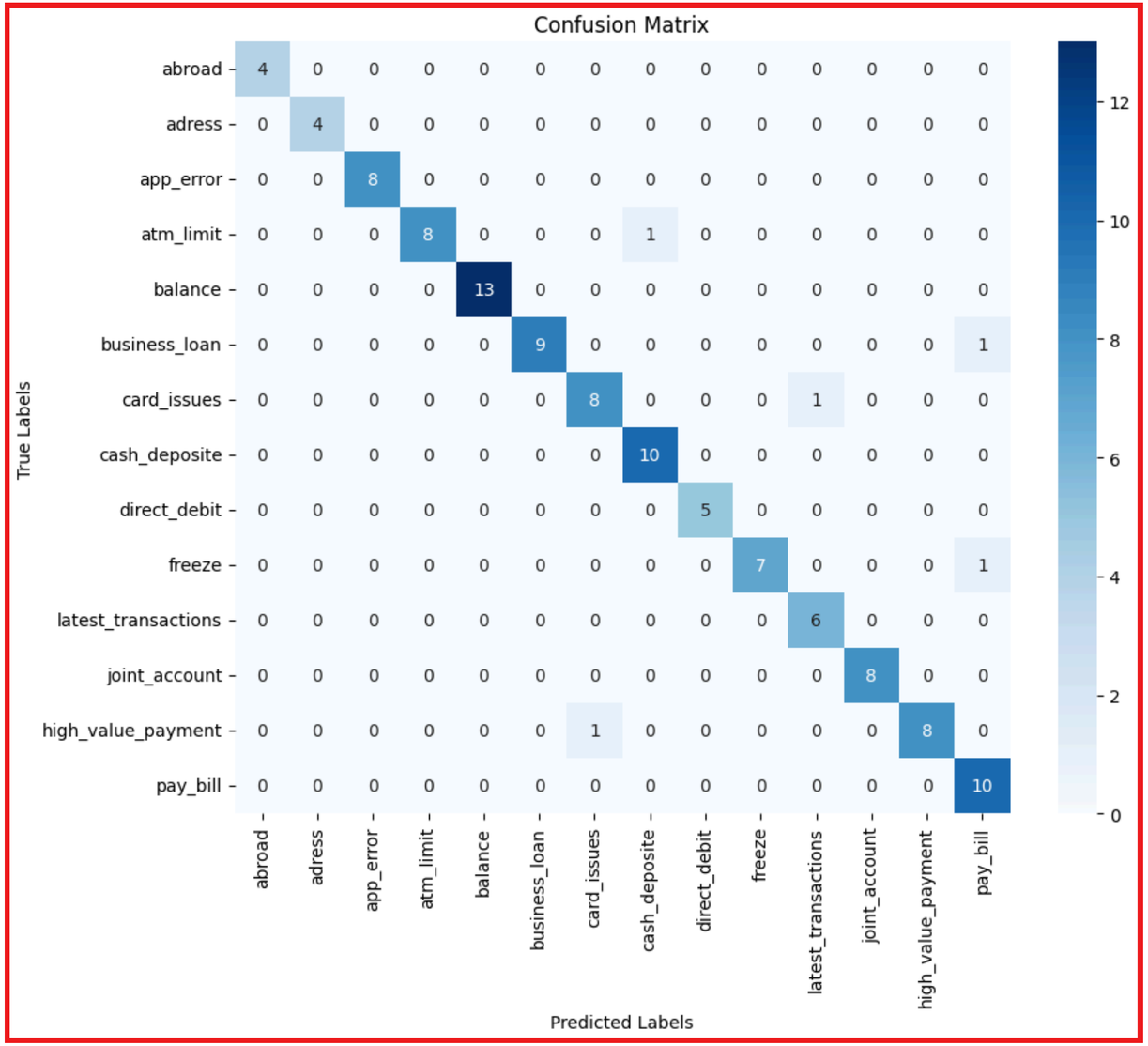
Fill-Mask • Updated Feb 19 • 20.4M • 426

```
model_name = "roberta-base"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(
        model_name, num_labels=14)
tokenized_train = train_dataset.map(tokenize_function, batched=True)
tokenized_test = test_dataset.map(tokenize_function, batched=True)

training_args = TrainingArguments(
    output_dir="./RoBERTa-IntentClassifier",
    evaluation_strategy="steps",
    per_device_train_batch_size=16, per_device_eval_batch_size=16,
    num_train_epochs=20, learning_rate=1e-5)
```

kairaamilanii/RoBERTa-minds14-en

Text Classification • Updated 6 days ago • 53

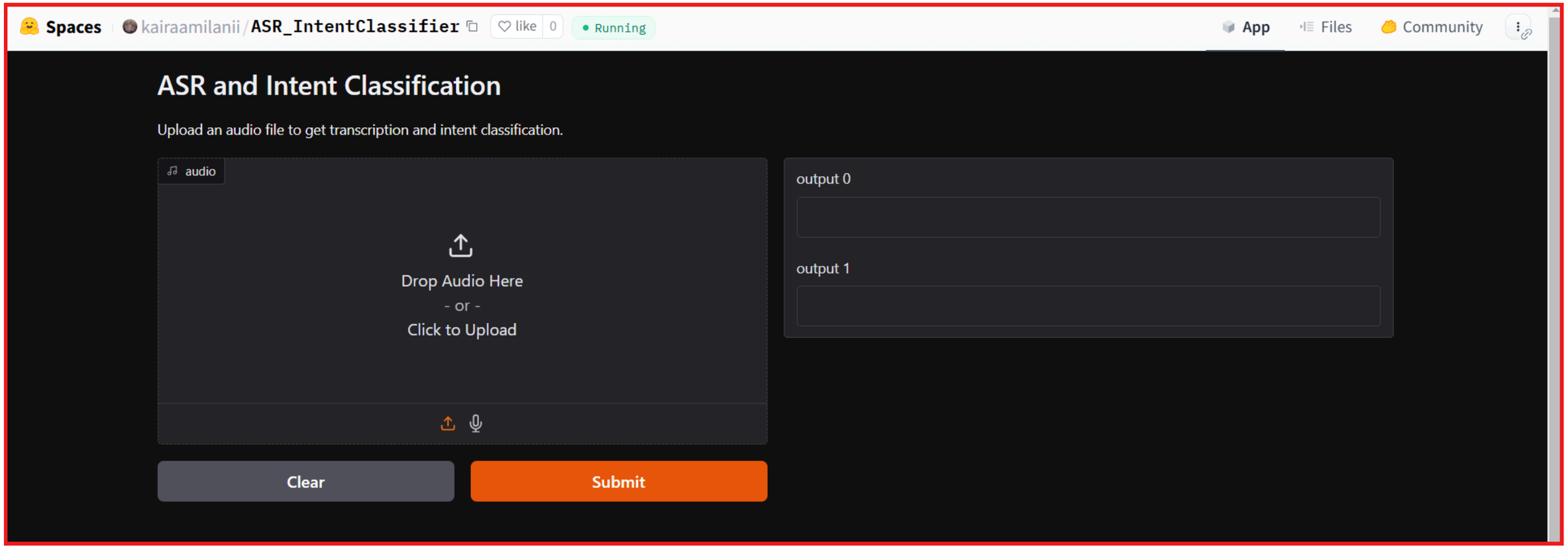
Accuracy	Precision	Recall	F1-Score
0.95	0.96	0.95	0.95



Deployment

```
model_asr = "kairaamilanii/whisper-mind14-enUS"
model_class = "kairaamilanii/RoBERTa-minds14-en"

transcriber = pipeline("automatic-speech-recognition", model=model_asr, chunk_length_s=30)
classifier = pipeline("text-classification", model=model_class)
```

```
def process_audio(audio):
    text_asr = transcriber(audio)['text']
    intent_class = classifier(text_asr)
    return text_asr, intent_class
```



Flow Work

Data Preparation

- .parquet dataset processing
  - audio (array)
  - transcription (str)
  - intent_class (int)
- EDA
  - Intent Distribution
- Cleaning Data
  - Remove null & duplicate row
  - Remove special characters

audio + transcript

transcript + intent class

ASR : Modeling

- Model + Tokenizer
  - openai / whisper-tiny
- Data Encoding
  - feature + label
- Training & Testing

Intent Classifier: Modeling

- Model + Tokenizer
  - FacebookAI / roberta-base
- Training & Testing

kairaamilanii / whisper-mind14-enUS

Accuracy + WER + F1-Score

kairaamilanii / RoBERTa-minds14-en

Accuracy + Precision + Recall + F1-Score

Deployment

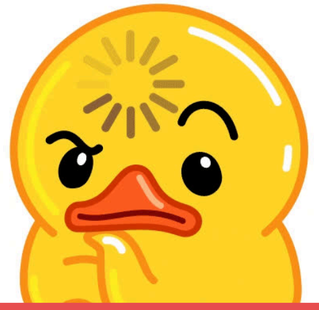
Classifier Model

ASR Model

configuration in “app.py”

Hugging Face + gradio

Future Improvement



Train the model for other variations of language, other topics dataset, and other form of audio sample files.

Investigate optimal hyperparameter settings, and apply noise reduction, speech enhancement and some techniques to improve transcription quality.

The large gap between validation and testing suggests overfitting. Use some techniques like cross-validation, dropout regularization in speech patterns, noise, and accents to improve performance

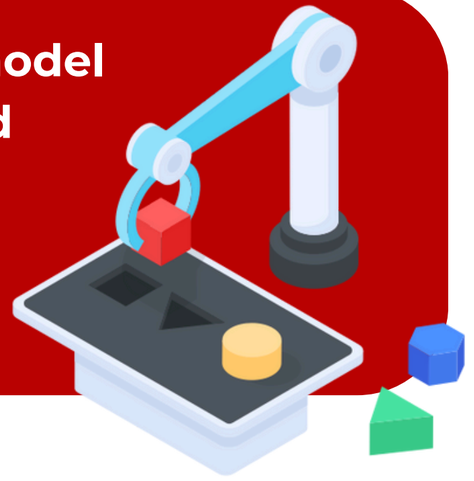
Conclusion

The trained Whisper model performs effectively in transcribing English audio with 81% accuracy

The RoBERTa-based classifier excels with an impressive accuracy, precision, recall, and F1-score of 0.95 across all metrics

The improvement for all metrics score means that training the model are effectively enhance model performance

Choosing the suitable model for each task are needed when build a model combination for final deployment



Thank You

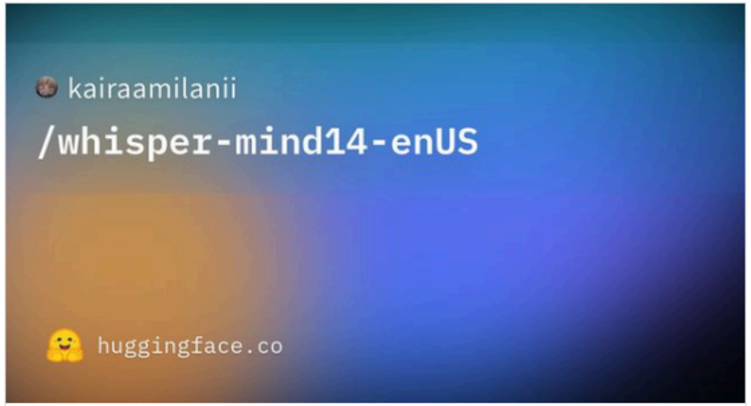
Running

ASR IntentClassifier

ASR and Intent Classifier based on minds-14 dataset.

Documentations:

kairamilanifitria/**NLP-Projects**



kairaamilanii/whisper-mind14-enUS · Hugging Face

We're on a journey to advance and democratize artificial intelligence through open source and open science.

huggingface

kairaamilanii/RoBERTa-minds14-en · Hugging Face

We're on a journey to advance and democratize artificial intelligence through open source and open science.

huggingface