Automatic Speech Recognition (ASR)

Speech-to-Text

Automatic Speech Recognition or ASR is the ability to consume a speech audio signal and output an accurate textual representation of said speech input.

Online vs Offline ASR

Online ASR happens as the speaker is speaking, the transcription process happening simultaneously with a minor lag.

Offline ASR on the other hand happens when the entire speech recording is available to us beforehand and our models can focus on transcribing the speech into its text as accurately as possible.

The ASR Pipeline

Broadly, the pipelines consist of two major components:

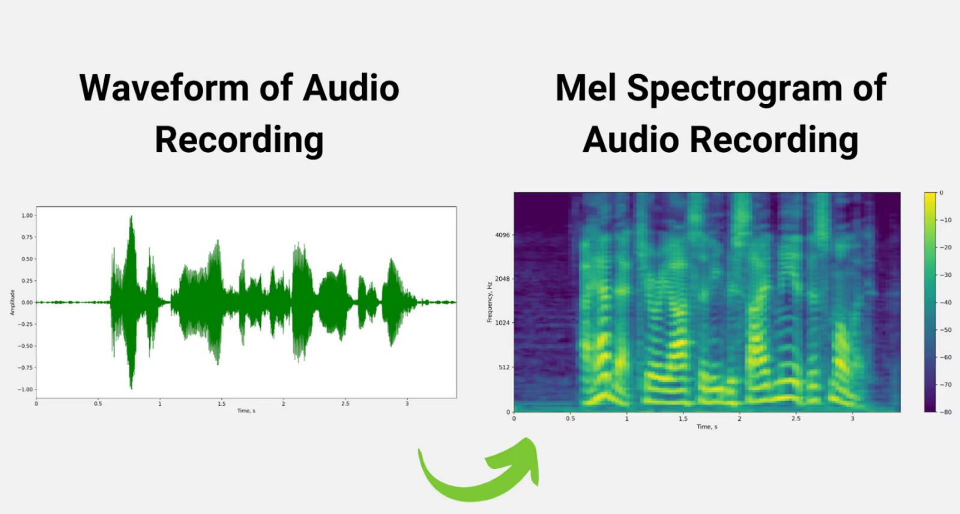
1. Acoustic encoders
2. Textual decoders

The acoustic encoders output the encoding for a speech input so that it can be consumed by the decoders to output meaningful textual representations of the encoding.

Traditional ASR

1. Acoustic Feature Extraction: Teaching Machines to "Hear"

The process begins by converting raw audio signals into numerical representations that machine learning models can interpret. Techniques like Mel-Frequency Cepstral Coefficients (MFCCs) and Mel Spectrograms analyze speech signals, focusing on frequencies most relevant to human hearing. These features act as a "fingerprint," capturing unique patterns that distinguish one sound from another.



2. Acoustic Modeling : Identifying Phonemes

Acoustic models convert numerical features into phonemes—the basic sound units that form words. This is achieved using Hidden Markov Models (HMMs) for sequential alignment and Gaussian Mixture Models (GMMs) to account for variations like accents and speaking styles.

Example: For "Hello," the model identifies phonemes as:
/h/ for the initial sound
/ɛ/ for the vowel
/l/ for the "l" sound
/oʊ/ for the final vowel sound

Forced Alignment: In training, predefined word boundaries help align phonemes accurately to their corresponding audio segments, ensuring precision.

3. Language Modeling: Predicting Word Sequences

Phonemes alone don’t convey meaning, so a Language Model predicts the most likely sequence of words based on context. Statistical models like N-grams or advanced neural networks evaluate probabilities of word combinations.

Example of Neural Networks:
Input: "I would like a cup of..."
Prediction: Likely completions include "coffee," "tea," or "water" based on context.

4. Pronunciation Modeling (Lexicon): Mapping Words to Sounds

A pronunciation dictionary bridges the gap between written text and its phonetic representation, ensuring the ASR system understands how words sound.

Example: For the word "cat," the lexicon provides the phonetic transcription /k/ /æ/ /t/.

5. Decoding: Combining Models for Transcription

The decoder integrates information from the acoustic model, language model, and lexicon to produce the most accurate transcription.

Example:
Input: "The quick brown fox jumps over the lazy dog."
Process:
Acoustic Model: Identifies phonemes.
Language Model: Predicts the most likely sequence of words.
Lexicon: Maps phonemes to their corresponding words.
Output: "The quick brown fox jumps over the lazy dog."

Downsides of the Traditional Hybrid Approach

- **Lower Accuracy**: Struggles with noisy or overlapping speech environments.
- **Complexity**: Requires the integration and fine-tuning of multiple independent components.
- **Data Dependency**: Relies heavily on forced-aligned data, which is costly and time-consuming to obtain.
- **Limited Adaptability**: Retraining is often necessary for new languages, accents, or domains, reducing flexibility.

End-to-End Deep Learning Approach

The modern approach to Automatic Speech Recognition (ASR) replaces the traditional multi-component architecture with a streamlined neural network. This single model directly maps acoustic features to text, eliminating the need for separate acoustic, pronunciation, and language models. The result is a simpler, more accurate, and adaptable system.

1. Acoustic Feature Extraction

As in the hybrid approach, raw audio signals are processed to extract relevant features, such as Mel-Frequency Cepstral Coefficients (MFCCs) or spectrograms. These features serve as the input to the deep learning model.

2. Deep Learning Model

A single neural network learns the mapping from acoustic features to text. Common architectures include:

- Connectionist Temporal Classification (CTC): Aligns speech and text sequences without requiring pre-aligned data, making it effective for variable-length input and output.
- Listen, Attend, and Spell (LAS): Utilizes attention mechanisms to focus on relevant parts of the audio, improving transcription accuracy.
- Recurrent Neural Network Transducer (RNN-T): Designed for real-time applications, it provides continuous, frame-by-frame recognition.

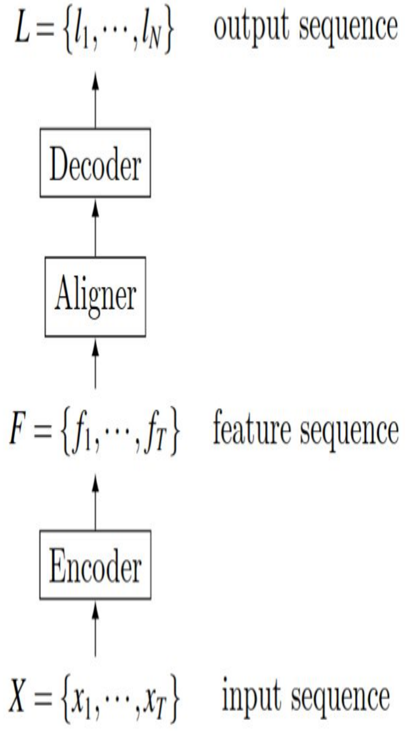

$L = \{l_1, \cdots, l_N\}$ output sequence
Decoder
Aligner
$F = \{f_1, \cdots, f_T\}$ feature sequence
Encoder
$X = \{x_1, \cdots, x_T\}$ input sequence

Figure 1. Function structure of end-to-end mode

Advantages of End-to-End ASR

- **Higher Accuracy**: End-to-end models outperform traditional systems, especially in noisy, conversational, or accented speech scenarios. Their ability to learn directly from raw data reduces errors associated with intermediate processing steps.
- **Simplified Training and Development**: Training a single model simplifies the pipeline, reducing the complexity and time needed for development. This approach eliminates the need for separate tuning of multiple components.
- **Adaptability**: End-to-end models are more adaptable to new languages, dialects, or domains. With sufficient training data, they can quickly generalize to new tasks and environments.
- **Continuous Improvement**: Advances in deep learning research, including innovations in architectures (e.g., Transformers) and optimization techniques, lead to ongoing improvements in accuracy, efficiency, and robustness.

Whisper for ASR

Whisper is a strongly **supervised** speech recognition model that eliminates the need for dataset-specific fine-tuning. It achieves exceptional performance by being pretrained on an expansive dataset of **680,000 hours** of **labelled audio**.

Key Features and Tasks

1. Extensive Multilingual Dataset

Whisper's dataset includes:

- **117,000 hours** of audio covering **96 languages**.
- **125,000 hours** of translation data for **X → English** tasks.

2. Supported Tasks

Whisper processes the same input audio signal to perform:

- **Transcription**: Converts speech into text in any language out of 96 languages.
- **Translation**: Translates speech in one language into text in English Language.
- **Voice Activity Detection (VAD)**: Identifies speech versus non-speech segments.
- **Alignment**: Aligns audio with text timestamps.
- **Language Identification**: Detects the language being spoken.

Model Architecture

Whisper is built on a state-of-the-art **encoder-decoder Transformer architecture** (Vaswani et al., 2017), known for its scalability and efficiency. Key architectural components include:

Input Processing

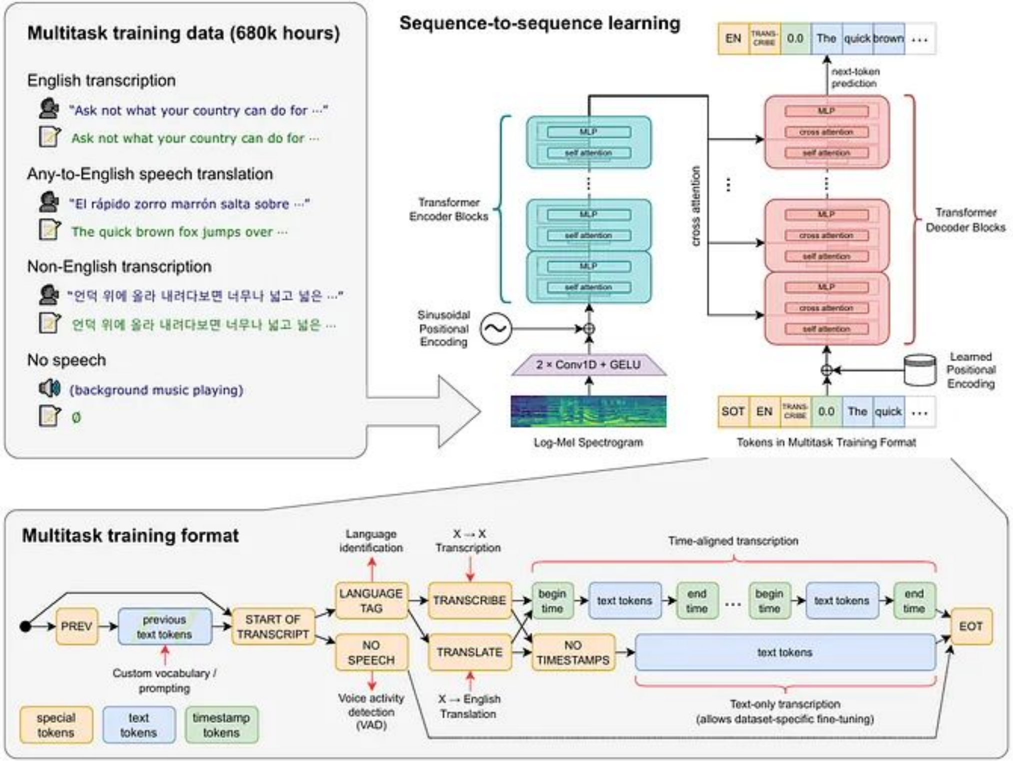
- Input audio files breaks into **30-second** segments paired with the subset of the transcript that occurs within that time segment.
- **Audio Representation**: All audio is re-sampled to **16,000 Hz**, and features are extracted using an **80-channel log-magnitude Mel spectrogram** representation is computed on 25-millisecond windows with a stride of 10 milliseconds.
- **Normalization**: Input features are scaled to range between -1 and 1 with zero mean across the pre-training dataset.

Encoder

- The encoder processes this input representation with a small stem consisting of two convolution layers
- Two convolution layers:
- First layer: Filter width of 3, using GELU activation.
- Second layer: Stride of 2.
- **Sinusoidal Position Embeddings**: Added to the convolution output before applying Transformer blocks.
- **Transformer Blocks**: Utilize pre-activation residual connections and layer normalization for effective learning.

Decoder

- Uses **Learned Position Embeddings** and tied input-output token representations.
- Uses **Byte-Level BPE Tokenizer** for the English only models and refit the vocabulary (but keep the same size) for the multilingual models to avoid excessive fragmentation on other languages since the GPT-2 BPE vocabulary is English only.


Multitask training data (680k hours)
English transcription
"Ask not what your country can do for ⋯"
Ask not what your country can do for ⋯
Any-to-English speech translation
"El rápido zorro marrón salta sobre ⋯"
The quick brown fox jumps over ⋯
Non-English transcription
"언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯"
언덕 위에 올라 내려다보면 너무나 넓고 넓은 ⋯
No speech
(background music playing)
∅
Sequence-to-sequence learning
EN
TRANS-CRIBE
0.0
The
quick
brown
⋯
next-token prediction
MLP
self attention
cross attention
Transformer Encoder Blocks
Transformer Decoder Blocks
cross attention
Sinusoidal Positional Encoding
2 × Conv1D + GELU
Log-Mel Spectrogram
Learned Positional Encoding
SOT
EN
TRANS-CRIBE
0.0
The
quick
⋯
Tokens in Multitask Training Format
Multitask training format
Language identification
X → X Transcription
Time-aligned transcription
PREV
previous text tokens
START OF TRANSCRIPT
LANGUAGE TAG
NO SPEECH
TRANSCRIBE
TRANSLATE
begin time
text tokens
end time
⋯
begin time
text tokens
end time
NO TIMESTAMPS
text tokens
EOT
Custom vocabulary / prompting
Voice activity detection (VAD)
X → English Translation
Text-only transcription (allows dataset-specific fine-tuning)
special tokens
text tokens
timestamp tokens

Evaluation Metrics

Word Error Rate (WER): Standard metric for speech recognition. Speech recognition systems are often evaluated using the Word Error Rate (WER), which measures differences between the model's output and a reference transcript based on string edit distance. However, WER penalizes even minor formatting differences, leading to higher scores even when transcripts are accurate by human judgment. This limitation affects all transcription systems.

Whisper achieves an **average WER of 12.8** for English speech recognition.