Automatic Speech Recognition

PAST, PRESENT, AND FUTURE

Contents

1 What is Automatic Speech Recognition?

Automatic Speech Recognition, or ASR, converts audio data into data formats that data scientists use to get actionable insights for business, industry and academia. It is a method to change unstructured data (data not organized in a pre-defined manner) into structured data (organized, machine readable, and searchable).

Most often, that converted data format is a readable transcript. Sounds simple, and—in principle—it is. Let’s unpack the three words behind ASR so we can make more sense of what is going on:

A

“Automatic” suggests that after a certain point, machines are doing some human task without any human intervention. Speech data in, and machine-readable data out.

S

“Speech” tells us that we are working with audio data. These can be noisy customer call recordings from an angry customer on a 16-lane highway in Los Angeles, to super-crisp, extra bass-y podcast audio, or anything in between.

R

“Recognition” tells us that our goal here is to convert the audio into a format that computers can understand (often a text transcript). In order to do neat things with audio data—such as trigger a command to buy something online (“Alexa, buy more toilet paper,”) or figure out what sort of phone sales interactions lead to better sales numbers—you need to convert audio data into a parsable data format for machines (and humans) to analyze.

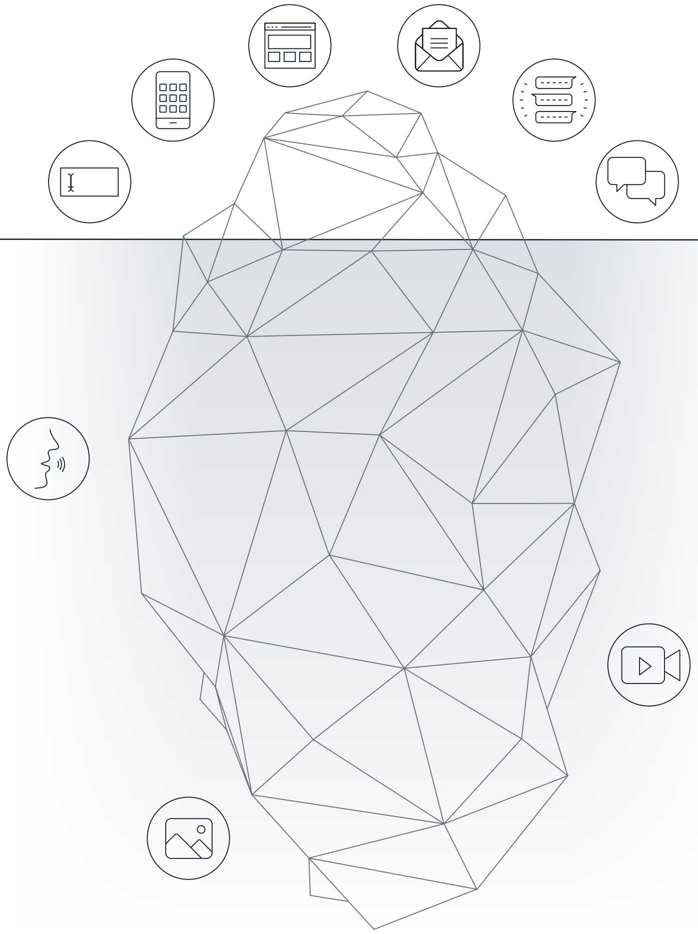
2

Why is Automatic Speech Recognition Important?

When you look at all the data being generated in the world, only 10% of that data is structured data.* That means 90% of the world's data is unstructured; unsearchable and unorganized. In addition, unstructured data is forecasted to increase by 60% per year. When you think about it, many organizations are **making important decisions on only 10% of the data.**

STRUCTURED DATA
Emails, forms, websites, apps, chat, and SMS

UNSTRUCTURED DATA
Videos, photos, and voice



Most of this unstructured data is voice or video data that needs to be changed into machine readable data to be used for decision-making. This is where ASR comes in and why it is important.

What could you do if you could data mine all your customer calls, sales conversations, retail engagements, patient conversations, and every physical, conference call and video meeting?

EXAMPLES OF CURRENT USES

CREATE VOICEBOTS / CONVERSATIONAL AI

IMPROVE CUSTOMER EXPERIENCE

REDUCE COMPLIANCE ISSUES

FIND IDEAS FOR NEW PRODUCTS OR SERVICES

BETTER ENABLE SUPPORT AGENTS IN REAL-TIME

MAKE SHOPPING EASIER

FIND EMPLOYEES OR PATIENTS WHO NEED SUPPORT

REDUCE CUSTOMER CHURN

The use cases are really endless, when you have access to structured voice data that is searchable and organizable. Search the data that is the true Voice of the Customer (VOC) or Voice of the Employee (VOE).

How does Automatic Speech Recognition Work?

ASR is a programmatic way to turn voice into text. Voices come in different dialects, languages, and with various levels of background noise. The objective of a good ASR is to turn the words from the voice or voices into the correct text. How is that done?

ASR was traditionally done in a series of seven steps:

TRADITIONAL METHOD (TRI-GRAM MODEL)

1 → LOAD AUDIO.

2 → REMOVE BACKGROUND NOISE.

3 → BREAK WORDS INTO PHONEMES OR UTTERANCES.

4 → STATISTICALLY GUESS THE SOUNDS.

5 → STATISTICALLY COMBINE THE SOUNDS TO GUESS THE WORDS.

6 → USE CONTEXT TO DECIDE ON THE BEST WORDS TO USE.

7 → OUTPUT A BEST GUESS.

Let's break each step down...

Where ASR Began

ASR technologies began development in the 1950 and 1960s, when researchers made hard-wired (vacuum tubes, resistors, transistors and solder) systems that could recognize individual words, not sentences or phrases. That technology, as you might imagine, is essentially obsolete.

The first known ASR was developed by Bell Labs and used a program called Audrey, which could transcribe simple numbers. The next breakthrough did not occur until mid-1970 when researchers started using Hidden Markov Models (HMM). HMM uses probability functions to determine the correct words to transcribe.

FULL HISTORY OF ASR

The important note on this traditional ASR process is that it is serial; **each step needs to be completed in order for the next step to be done.** Think of it like an assembly line.

1 First, you load the audio into the ASR process. It can be pre-recorded audio (batch mode) or real-time live audio (streaming). Depending on the ASR process, you may need to break up the audio into chunks of audio to feed into the process.

2 ASR then de-noises the audio to make sure it is free of dogs barking or garbage trucks in the background. De-noising audio can be tricky because you don't want to accidentally remove the voice from the audio. There are a variety of approaches here, but primarily speech recognition software is trying to limit the audio to just what falls into the range of human voice. Once it has an idea of where the voice is in the audio, it cuts that out and moves it onto the next step in the process.

3 Now that the voice is extracted from the audio, the ASR process needs to know what sounds make up different words. Think about how each letter makes a specific noise, and how some combinations, like "ph," are unique noises. These sounds that make up words are called phonemes. The voice data is sliced into individual phonemes, which then make it easier for the software to put them together as words.

4 The words that an automatic speech recognition can identify comes from its programmed dictionary. Each phoneme is guessed using a statistical model; i.e. it is more likely the sound is "ai" than "oi".

5 Then you take these phonemes and put them together to create guessed words. These guessed words are statistically based similar to the phonemes; i.e. how likely would these sounds make this word.

6 "Pair" and "pear" sound the same, but are different words. How do you know which word is being said? You can use the context around the word to identify what it means. If I am asking you for a "pair of headphones," you understand

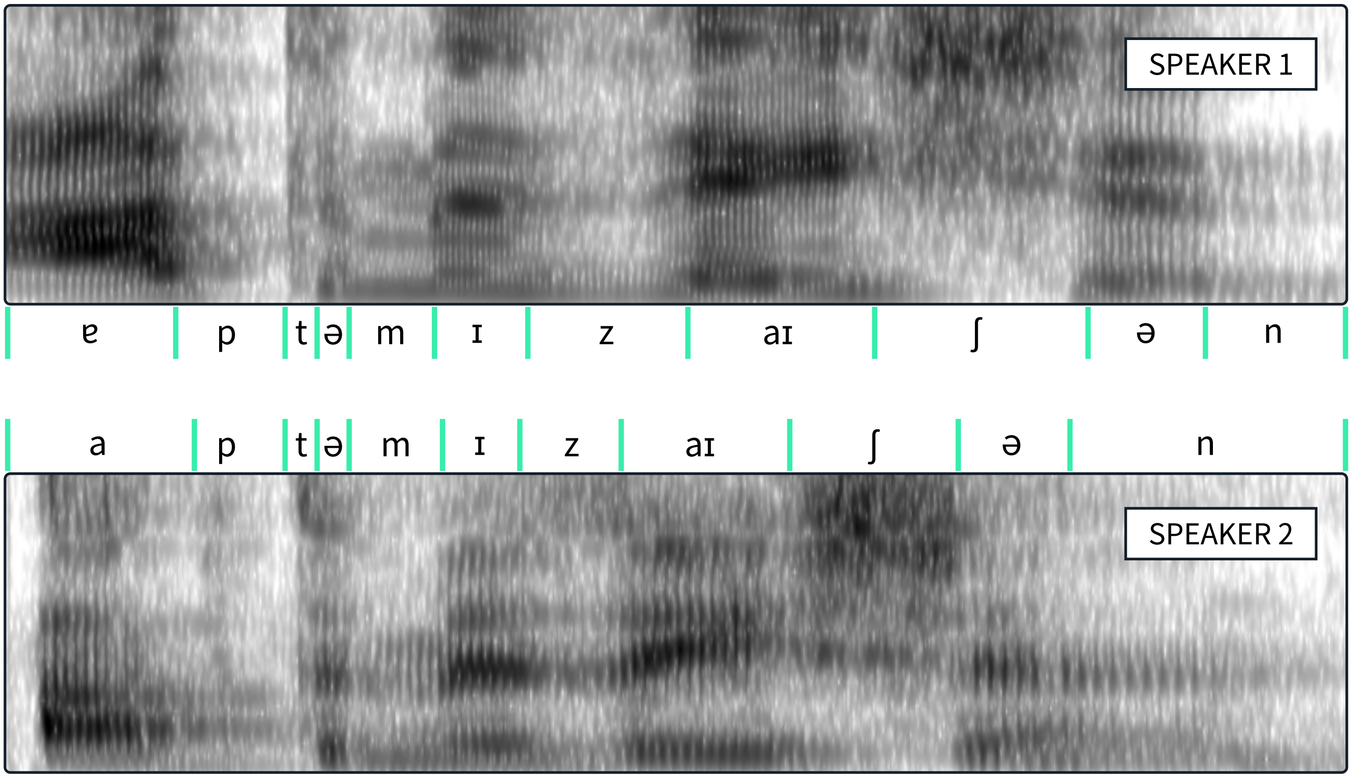
from the context which “pair” is being used. ASR uses the same trick to identify which word is being spoken. As more of the text is output from the phonemes, an ASR process might go back and correct itself. That way, the final results you are provided are better.

7 Finally, the traditional process outputs its best guess as a transcript.

Why ASR is Hard...Everyone Speaks Differently

Between any two speakers there are variations in pronunciation, tone, word-choice, grammar choice, even amount of lung pressure, that from a mathematical perspective (computers speak math) make what they say completely different; even if it sounds the same to you and me. In fact, even if the same person utters a sentence twice, the sounds when recorded and measured are mathematically different.

These are two spectrograms of **two people saying the same word: optimization**. Spectrograms are one way to visualize audio data. As you can see, these two spectrograms are very different from one another. Pay special attention to the darker lines and their relative shapes. Same word to our human brains, but two mathematical realities for computer brains.



HYBRID METHOD (TRI-GRAM PLUS NEURAL NETWORKS/ARTIFICIAL INTELLIGENCE)

The traditional method worked well for low noise, single speaker environments, but it did not work so well for multi-speaker and noisy environments. Luckily, in the late 1980's artificial intelligence and neural networks were starting to be studied for ASR but they were not truly added to this traditional method til mid-2000s.

This hybrid method still uses the seven step serial process but added neural networks that could learn from the audio data and human transcriptions to better denoise the audio, improve on phoneme identification. AI was also added for post transcription processing to fix words out of context or correct keywords. Unfortunately, this adds yet another step to the traditional process.

HYBRID METHOD PROS

- Better denoise audio
- Improve on phoneme identification
- Enables post-transcription corrections

HYBRID METHOD CONS

- Adds another step to the process—i.e. more time
- Requires more computing power — i.e. more resources and costs
- Generalized, not specific to customer needs and can't be trained to improve

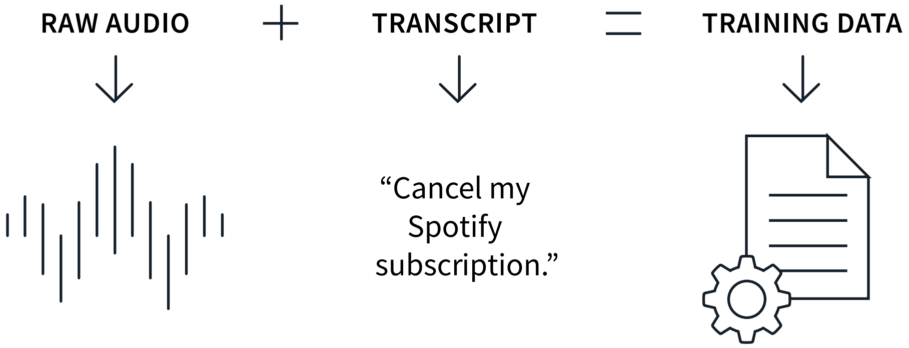
The downside to this hybrid method is that it is still a serial step-by-step process but now more complex and computational resource hungry; it uses a lot of computing power. Imagine having an old cookie assembly line, but now you have added robots to make sure the exact ingredients and portions are put in. You also added an additional quality assurance step for robots to pick out the badly shaped cookies. Now you can make more precise shaped and same tasting cookies but the assembly line is still the old one.

Traditional ASR systems built using the hybrid method are designed to deal with "general" audio, not specialized audio for industry, business, or even academic purposes. In other words, they provide generalized speech recognition and cannot realistically be trained to improve on your speech data.

END-TO-END DEEP LEARNING METHOD (ALL NEURAL NETWORKS/ARTIFICIAL INTELLIGENCE)

In the 2010s, researchers believed that neural networks were the key to having a new type of ASR. With the advent of big data, faster computers, and graphical processing unit (GPU) processing, a new ASR method was developed, End to End Deep Learning ASR. This new ASR method could “learn” and be “trained” to become more accurate as more data is fed into the neural networks.

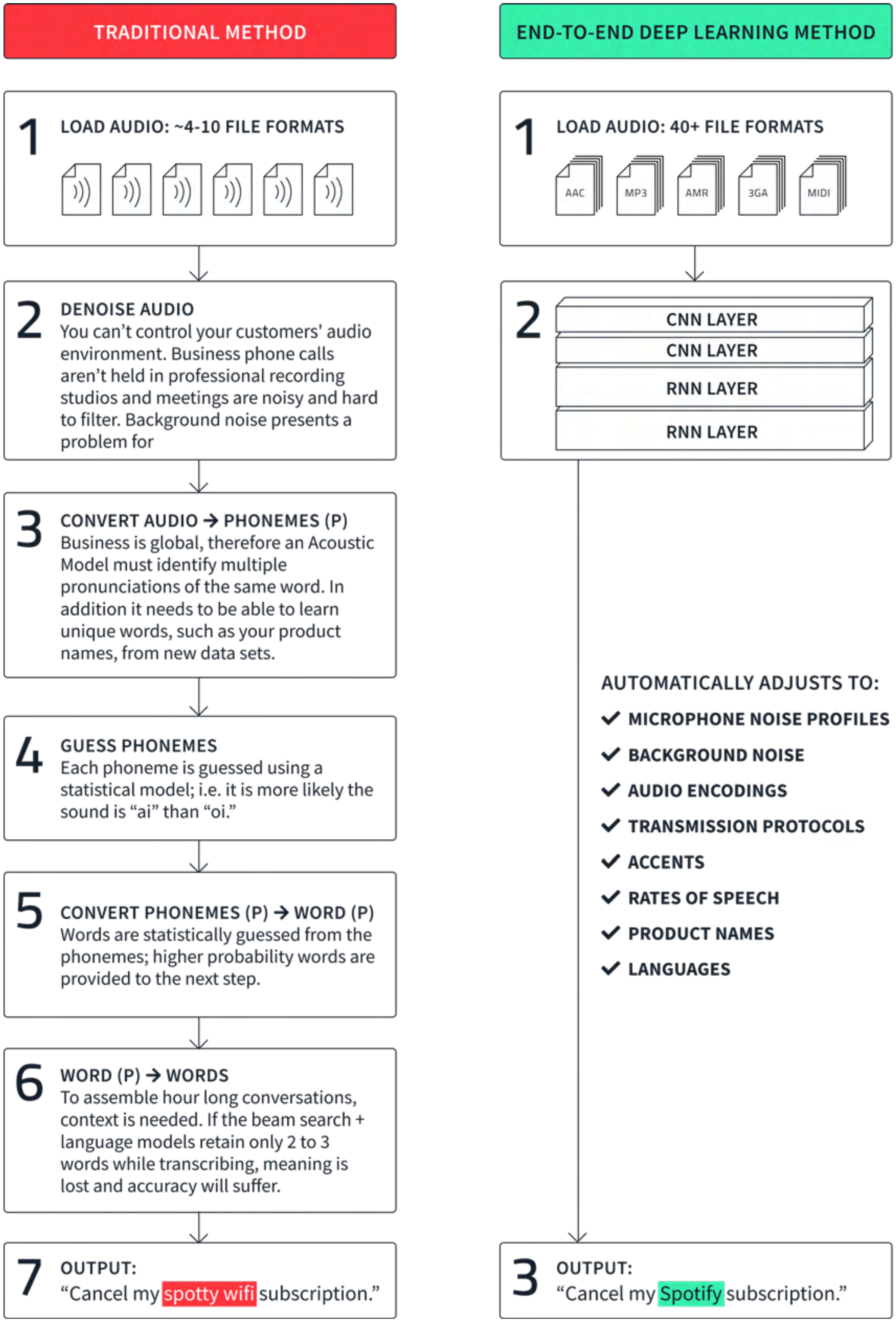
Unlike traditional ASR systems, which are trained by meticulously editing sub-components of a data pipeline, an End-to-End Deep Learning Neural Network improves with each data set it receives. You can continuously train your model with the voice of your customers, and it will improve identification of sounds, and subsequently the words in new audio submitted.



With the End-to-End Deep Learning method, we can build ASR systems that allow us to make highly accurate transcripts of audio data that have specialized words, accents, noise conditions etc., but with a technology that scales well and is always evolving.

For example, modern ASR systems are able to further overcome noise and speaker variability issues by taking linguistic context into account. This means that computers are learning to distinguish the meaning of the word “tree” when found in conversations about family versus ones about botany.

COMPARISON OF TRADITIONAL VS. END-TO-END DEEP LEARNING ASR METHODS



TRADITIONAL METHOD

1 **LOAD AUDIO: ~4-10 FILE FORMATS**

2 **DENOISE AUDIO**
You can’t control your customers' audio environment. Business phone calls aren’t held in professional recording studios and meetings are noisy and hard to filter. Background noise presents a problem for

3 **CONVERT AUDIO → PHONEMES (P)**
Business is global, therefore an Acoustic Model must identify multiple pronunciations of the same word. In addition it needs to be able to learn unique words, such as your product names, from new data sets.

4 **GUESS PHONEMES**
Each phoneme is guessed using a statistical model; i.e. it is more likely the sound is “ai” than “oi.”

5 **CONVERT PHONEMES (P) → WORD (P)**
Words are statistically guessed from the phonemes; higher probability words are provided to the next step.

6 **WORD (P) → WORDS**
To assemble hour long conversations, context is needed. If the beam search + language models retain only 2 to 3 words while transcribing, meaning is lost and accuracy will suffer.

7 **OUTPUT:**
“Cancel my spotty wifi subscription.”

END-TO-END DEEP LEARNING METHOD

1 **LOAD AUDIO: 40+ FILE FORMATS**

2

AUTOMATICALLY ADJUSTS TO:

- ✓ MICROPHONE NOISE PROFILES
- ✓ BACKGROUND NOISE
- ✓ AUDIO ENCODINGS
- ✓ TRANSMISSION PROTOCOLS
- ✓ ACCENTS
- ✓ RATES OF SPEECH
- ✓ PRODUCT NAMES
- ✓ LANGUAGES

3 **OUTPUT:**
“Cancel my Spotify subscription.”

What is the future of Automatic Speech Recognition? 4

The ultimate ASR would be one solution that:

- Can transcript any language, accent, and dialect
- Can denoise any audio stream, and pick out all conversations for transcription in milliseconds
- Knows when you are switching between languages and can translate in that language
- Can tell you directly who is speaking with saved voice patterns and what mood they are in

The unified ASR solution! We are not quite there yet.

But, when you look at the current ASR methods, you can see that End-to-End Deep Learning is the only method that can possibly get you to that unified ASR solution because only End-to-End Deep Learning can actually learn and improve with data. The more data you have, the better the ASR gets.

Conclusion 5

We hope this guide has given you a solid understanding of what Speech Recognition is as well the differences between types of ASRs. This knowledge can help you see the potential power in having access to more accurate, more usable voice data.

Can you see a time when you are having human-like conversations with a voicebot and don’t even know it? Or be able to get immediate assistance sent to you because of the concern or fear in your voice? Or even design new inventions by just talking — like “Jarvis” in the Iron Man movies?

At Deepgram, we see and are working toward this future. If you’re ready to build the next great voice product or to start using voice data as a growth or cost-savings strategy instead of just checking the compliance box, we’re here to help you make it happen.

Deepgram: no compromises, only opportunities.

ADDITIONAL RESOURCES AND SUGGESTED LINKS:

Create a Free Account

Developer Documentation

State of ASR Report

How to Vet an ASR Provider

Contact Us