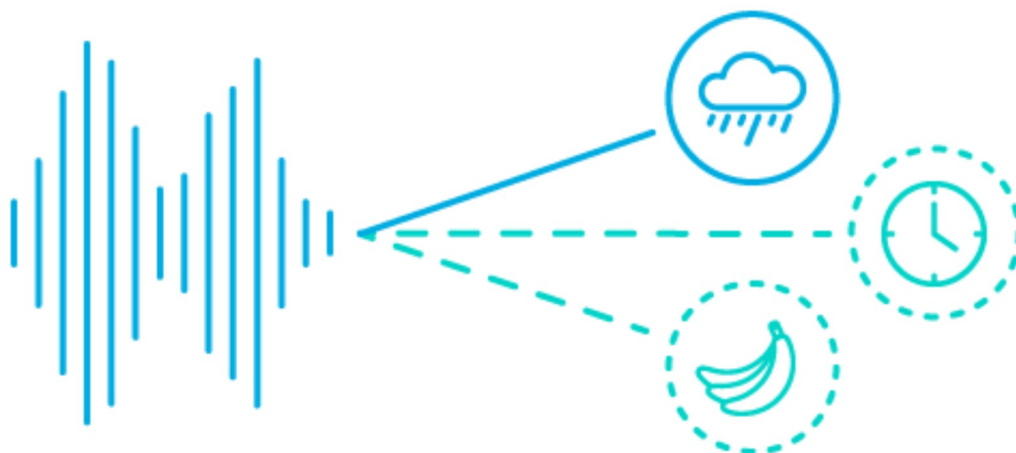


What Is Natural Language Understanding (NLU)?

NLU: Building Smart Computers

With natural language understanding (NLU), computers can deduce what a speaker actually means, and not just the words they say. In short, it is what enables voice technology like Alexa to infer that you're probably asking for a local weather forecast when you ask, "Alexa, what's it like outside?"

Today's voice-first technologies are built with NLU, which is artificial intelligence centered on recognizing patterns and meaning within human language. When a computer understands what you mean to say without you having to ask it in one specific way, using your voice starts to feel like having an actual conversation.



Teaching Computers to Interpret with Voice

Communication is a constant exercise in deciphering meaning; sometimes we use the wrong words, and often the words we say are not actually the words we mean. NLU is all about providing computers with the necessary context behind what we say, and the flexibility to understand the many variations in how we might say identical things.

Before NLU, designing a weather app with voice as an input would require a list of a thousand ways we could ask "is it raining." With NLU, Alexa-enabled devices like Amazon Echo can apply learnings from historical interactions, across thousands of diverse applications, to understand that "is it raining outside" and "is it going to rain" are essentially the same question. This flexibility allows voice experiences to offer a faster, easier, and more delightful way of interacting/engaging with technology.

Voice technologies that incorporate NLU allow developers to focus more on designing useful voice experiences, and less on trying to infer what a user is trying to say. Here are four tips to designing natural voice-first experiences:



1. Identify Intent

What are the different goals someone wants to accomplish? Be specific.

2. Identify Utterances

What are the different words or phrases people might say to signal their goal and intent? The more examples, the better.

3. Cover Corrections

Natural conversation isn't perfect. Give users the opportunity to correct errors or change their answers.

4. Build Exceptions

It's always better to say "I don't know the answer to that," than to pretend and give a wrong answer.

There are many other layers to voice design. For example, the voice user interface should be concise and present only as much information as needed. Like a natural conversation, progressively build on a user's response with additional information to move the user towards their goal.

Learn more in the [Amazon Alexa Voice Design Guide](#).

Build for Voice Today

How can you start taking advantage of NLU to build for voice? You can start creating innovative voice experiences today with the Alexa Skills Kit (ASK), which leverages Amazon's knowledge and pioneering work in the field of voice design. ASK is a collection of self-service APIs, tools, documentations, and code samples that makes it fast and easy to build for Alexa—the cloud-based voice service available on tens of millions of devices.

Design for voice today and shape the technology of the future.



[Back to top](#)

Alexa Skills Kit

Alexa Skills Kit

Learn

Design

Build

Launch

Resources

Quick Start Guide

Documentation

Developer Forums

Agencies and Tools

Alexa Voice Service

Alexa Voice Service

Learn

Design

Build

Launch

Resources

Quick Start Guide

AVS Device SDK

AVS API

Dev Kits for AVS

Connected Devices

Alexa Smart Home

Alexa Gadgets

Agreements

Amazon Developer Services Agreement

Program Materials License Agreement

Mobile Ad Network Publisher Agreement

Mobile Ad Network Program Participation Requirements

Advertise Your App with Amazon Agreement

Trademark Guidelines

Terms of Use

AVS Program Requirements

Blogs

Alexa Developer Blog

AWS Blog

Support

Amazon Developer Support

Contact Us

Forums

Manage Email Preferences

Devices

Echo Show & Echo Spot

Amazon Echo, Echo Dot & Echo Plus

Amazon Tap

Amazon Fire TV

Language

English

Follow Us:

