

Conversational Language Understanding (CLU) using Microsoft Azure AI Language

2025

Conversational Language Understanding (CLU) with Azure Cognitive Services: A Conceptual Application Study

Project Overview

This project delivers a comprehensive exploratory analysis of Microsoft Azure's **Conversational Language Understanding (CLU)** service. As a core component of Azure Cognitive Services for Language, CLU empowers applications to interpret natural language by identifying user *intents* and extracting key *entities*. This study investigates the underlying CLU architecture, examines its text-processing pipeline, and conceptually applies it to a real-world domain—all without requiring active system deployment. The findings are documented in a structured report accompanied by interface screenshots from Azure Language Studio.

Business Problem

Modern enterprises process thousands of unstructured customer messages daily across chat, email, and social media. Manually triaging this text to determine specific user goals—such as booking a ticket, modifying an order, or retrieving policy information—is labor-intensive, error-prone, and delays response times. Conversational Language Understanding directly addresses this by automating the extraction of structured data from free-form language. This enables downstream systems to trigger appropriate business actions instantly, reducing operational overhead while significantly improving customer satisfaction and scalability.

Project Objectives

The primary objectives of this conceptual study were to explore the official Microsoft CLU documentation and interface via Azure Language Studio, analyze the complete processing pipeline—from text ingestion and tokenization to intent classification and entity tagging—and select a relevant business domain (Travel Booking) to illustrate how CLU can transform user utterances into actionable insights.

Solution Approach

The project began with an in-depth review of Azure's CLU documentation, followed by hands-on navigation of the Language Studio environment. The Travel Booking domain was selected to contextualize the technology. Representative user queries—such as "Book a flight to Paris next Friday" and "Change my hotel reservation"—were mapped to specific intents (e.g., Book Flight, Modify Reservation) with corresponding entity extraction (e.g., Destination, Date, Reservation Type). A conceptual testing strategy was outlined, covering accuracy metrics, confidence thresholding, and fallback mechanisms for out-of-scope inputs to ensure robust performance evaluation.



CHIRAG ARUNKUMAR VYAS

Key Findings

Strengths of Azure CLU include its no-code labeling interface, which allows domain experts to train models without machine learning expertise; prebuilt entity support for dates, locations, and numbers; and active learning features that accelerate model refinement. However, **limitations** were also identified: model accuracy is heavily dependent on the volume and diversity of training data, the system lacks native multi-turn conversation memory, and cloud inference introduces slight latency, which may challenge real-time interactive applications.

Technology Stack & Deliverables

The technologies explored include Azure Cognitive Services for Language and Azure Language Studio. The deliverables consist of a structured PDF report detailing the CLU pipeline and domain-specific application, along with curated screenshots from the Azure interface showcasing intent definition, entity tagging, and model training views.

Business Applications

This conceptual framework is directly applicable across multiple industries, including travel and hospitality for flight and hotel bookings, retail for product inquiries and order tracking, healthcare for appointment scheduling and medication FAQs, and banking for fund transfers and dispute resolution.

Table of content

Sr no.	Content	Page no.
2.1	Task Instructions	
	2.1 Understanding CLU in Azure.	
2.1.1	• Read through the Microsoft CLU Documentation. • Overview • What is conversational language understanding? • Workflow for Building a CLU Model	1 2 3
2.1.2	• Access Azure Language Studio and navigate to the Conversational Language Understanding section. Prerequisites. • An Azure subscription and its methods: • An Azure AI Language resource. • Create language resources. • Create a conversation project. • Importing and exporting project.	4 5 6
2.1.3	• Explore how CLU processes text inputs and recognizes user intents and entities. • Build Schemas • Guidelines and recommendations • Add intents • Add entities • Add prebuild component • Add list components • Define entity options	7 8
2.2	2.2 Exploring CLU Applications	
2.2.1	• Selecting a real-world use case where CLU could be applied: • Retail Chatbot – Assisting customers in finding products. • Healthcare Assistant – Answering patient inquiries. • Travel Booking System – Helping users with reservations.	
	• Universal Steps to Build CLU Use Cases Across Any Domain with MSA (Microsoft Azure) • Utterance and label your data • Creating Schemas • Label the utterances in Language Studio • Data labelling guidelines • How to label your utterances • Suggest utterances with Azure OpenAI • Add required configurations to Azure OpenAI resource • Data balancing • Data splitting • Training models • Deploy a model	9 10 11 12 13 14 15 17
2.2.2	• Chatbot use cases in Retail Chatbot – Assisting customers in finding products. • Chatbots answer questions and inquiries • Chatbots for Event Ticket Booking • Chatbots for Product Discovery, Inventory Checks, and Personalized Recommendations • Chatbots for Seamless Customer Experiences • Chatbots can process return and exchange requests • Chatbots can confirm orders and track shipping • Chatbots for Effortless Feedback Collection • Chatbots assign customer requests to support teams & Smart Ticket Routing with Chatbots • Chatbots Help Simplify Pricing – Key Notes • Chatbots for Engaging Product Promotion • Chatbots for Smarter Lead Generation • Chatbots for Quizzes, Promotions & Contests • Chatbots for Personalized Product Discovery • Chatbots for Strategic Upselling & Cross-Selling • Chatbots for Guided Product Discovery • Chatbots as 24/7 Personal Shopping Assistants • Chatbots as 24/7 Sales Agents • Chatbots for Instant Quotes & Estimates • references	18 19 20 21 22 23 24 25 26 27

2 Task Instructions

2.1 Understanding CLU in Azure

2.1.1 Read through the Microsoft CLU Documentation.

• Overview

Conversational Language Understanding (CLU) is part of the **Azure AI Language Service**.

It's a collection of natural language processing capabilities within **Azure Cognitive Services**, including:

- **Conversational Language Understanding (CLU)** – to detect **intents** and extract **entities** from natural language input.
- **Text Analytics** – for sentiment analysis, key phrase extraction, named entity recognition.
- **Question Answering** – to build knowledge-based Q&A systems.
- **Text Summarization and Translation** – to summarize or translate text. So, CLU is one **capability** within the broader **Azure Language Service**.
- Access Azure Language Studio and navigate to the Conversational Language Understanding section.
- Explore how CLU processes text inputs and recognizes user intents and entities.

Term	Description	Status
Language Understanding (LUIS)	Older service by Microsoft for understanding intents and entities from natural language.	Being retired (deprecated)
Conversational Language Understanding (CLU)	The new, improved version of LUIS, part of the Azure AI Language Service.	Current and supported

What's the Difference?

- LUIS (Language Understanding Intelligent Service) was the original service. It is being phased out and will be less operational
- CLU is the replacement for LUIS. It offers better performance, is part of Azure Language Service, and is tightly integrated with Language Studio.

• What is conversational language understanding?

Conversational Language Understanding (CLU) - Notes

- CLU is one of the custom features offered by Azure AI Language.
- It is a cloud-based API service that uses machine learning to enable building natural language understanding components.
- Designed for end-to-end conversational applications.

Key Features:

- Intention Detection: Predicts the overall intent of an incoming user utterance.
- Entity Extraction: Identifies and extracts important information (entities) from the text.
- Intelligence Only: CLU understands input text, but it does not perform any actions.
- Custom Model Development:
 - Developers can create a CLU project.
 - Label utterances (examples of input text).
 - Train and evaluate the model.
 - Publish when ready for use.

Important Notes:

- Model quality highly depends on the quality of labelled data.
- CLU provides a custom web portal for building and managing models.
- The web interface is available via Language Studio, which simplifies the development process.
- You can follow quickstart steps to begin creating your own model.

Example usage scenarios

CLU can be used in multiple scenarios across a variety of industries. Some examples are:

End-to-end conversational bot

Use CLU to build and train a custom natural language understanding model based on a specific domain and the expected users' utterances. Integrate it with any end-to-end conversational bot so that it can process and analyze incoming text in real time to identify the intention of the text and extract important information from it. Have the bot perform the desired action based on the intention and extracted information. An example would be a customized retail bot for online shopping or food ordering.

Human assistant bots

One example of a human assistant bot is to help staff improve customer engagements by triaging customer queries and assigning them to the appropriate support engineer. Another example would be a human resources bot in an enterprise that allows employees to communicate in natural language and receive guidance based on the query.

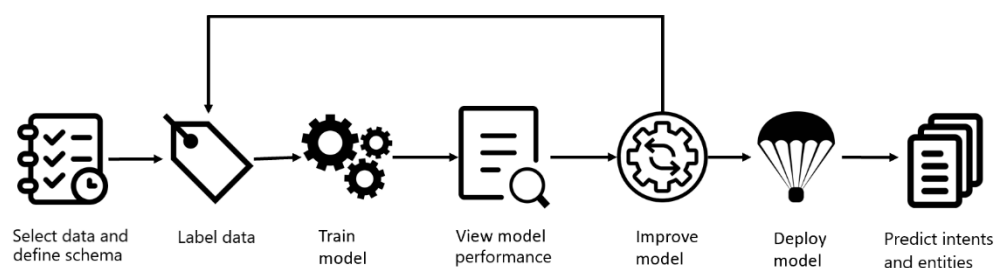
Command and control application

When you integrate a client application with a speech to text component, users can speak a command in natural language for CLU to process, identify intent, and extract information from the text for the client application to perform an action. This use case has many applications, such as to stop, play, forward, and rewind a song or turn lights on or off.

Enterprise chat bot

In a large corporation, an enterprise chat bot may handle a variety of employee affairs. It might handle frequently asked questions served by a custom question answering knowledge base, a calendar specific skill served by conversational language understanding, and an interview feedback skill served by LUIS. Use Orchestration workflow to connect all these skills together and appropriately route the incoming requests to the correct service.

Project development lifecycle Creating a CLU project typically involves several different steps.



• Workflow for Building a CLU Model

1. Define your schema: Know your data and define the actions and relevant information that needs to be recognized from user's input utterances. In this step you create the intents that you want to assign to user's utterances, and the relevant entities you want extracted.
2. Label your data: The quality of data labelling is a key factor in determining model performance.
3. Train the model: Your model starts learning from your labelled data.
4. View the model's performance: View the evaluation details for your model to determine how well it performs when introduced to new data.
5. Improve the model: After reviewing the model's performance, you can then learn how you can improve the model.
6. Deploy the model: Deploying a model makes it available for use via the Runtime API
7. Predict intents and entities: Use your custom model to predict intents and entities from user's utterances.

2.1.2 Access Azure Language Studio and navigate to the Conversational Language Understanding section.

- To access Azure language studio, we have to follow steps

• Prerequisites

Before you start using CLU, you will need several things:

- A) An Azure subscription
- B) An Azure AI Language resource

A. An Azure subscription:

Subscription Type	Best For	Cost	Support Level
Free Trial	Learning & testing	Free (limited time)	Basic
PAYG	Flexibility, startups	Based on usage	Moderate
EA	Large enterprise	Prepaid + discounts	Full MS support
MCA	Mid-sized business	Flexible billing	Moderate
Student	Verified students	Free (limited)	Basic
CSP	Businesses via partners	Varies	Depends on partner

Azure offers **different types of subscriptions** to give users flexibility based on their needs—whether for individual learning, enterprise use, or large-scale solutions. Here's a breakdown of the **main types of Azure subscriptions**:

1. Free Account / Azure Free Trial

- **Target Audience:** Students, individual learners, small experiments.

- **Includes:**

- **\$200 credit** for 30 days.
- Free access to **popular services** (e.g., Azure VMs, AI, Cosmos DB) for **12 months**.

○ Always-free limited access to some services beyond 12 months.

- Great for: Exploring, testing, small projects.

2. Pay-As-You-Go (PAYG) Subscription

- **Billing:** Based on usage — no upfront costs.
- **Who's it for?** Individuals, small businesses, developers needing flexibility.
- **You pay only for what you use** (billed monthly).
- Can scale up or down easily.

3. Enterprise Agreement (EA) Subscription

- For large organizations with **long-term commitments**.
- Offers **discounted pricing** and consolidated billing.
- Good for managing **multiple departments or teams** under one master billing account.
- Requires a **minimum financial commitment** (usually starts around \$100,000/year).

4. Microsoft Customer Agreement (MCA)

- Replaces older subscription models (like EA for small-medium businesses).
- Gives **more self-service control**.
- Direct billing through Microsoft or a partner.
- Ideal for medium-sized organizations that want **flexibility with governance**.

5. Azure for Students

- Free Azure account for **verified students**.
- Includes **\$100 credit** (renewable annually).
- No credit card needed to sign up.
- Access to selected Azure services like CLU, Functions, App Services.

6. Cloud Solution Provider (CSP)

- Managed by **Microsoft partners**.
- Useful if you want to **buy Azure services through a third party** who also provides support and billing.
- Good for businesses wanting **reseller support** or bundled services.

B. An Azure AI Language resource

An Azure AI Language resource is a cloud-based resource within Microsoft Azure that gives you access to powerful Natural Language Processing (NLP) capabilities. It powers services like:

- Conversational Language Understanding (CLU)
- Text Analytics
- Question Answering
- Named Entity Recognition
- Language Detection
- Sentiment Analysis

This resource acts as the backend brain for any application that needs to understand and process human language.

Key Use Cases

1. Building Chatbots with CLU (intent + entity recognition)
2. Analysing Text (e.g., sentiment analysis, key phrase extraction)
3. Creating Q&A bots from documents or knowledge bases
4. Language detection and translation
5. Extracting structured info from unstructured text (e.g., locations, dates, names)

• Create language resources

A. How to Create an Azure AI Language Resource

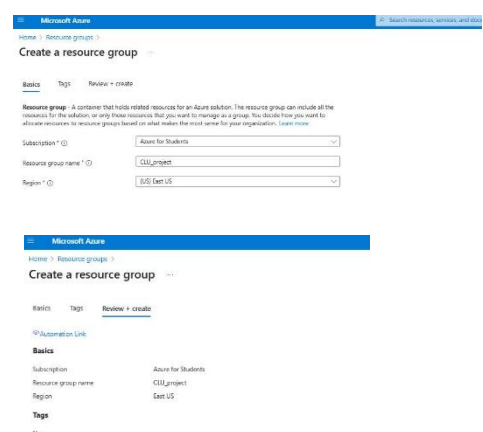
You can create it through the Azure Portal, Azure CLI, or via ARM templates. Here's the easiest method:

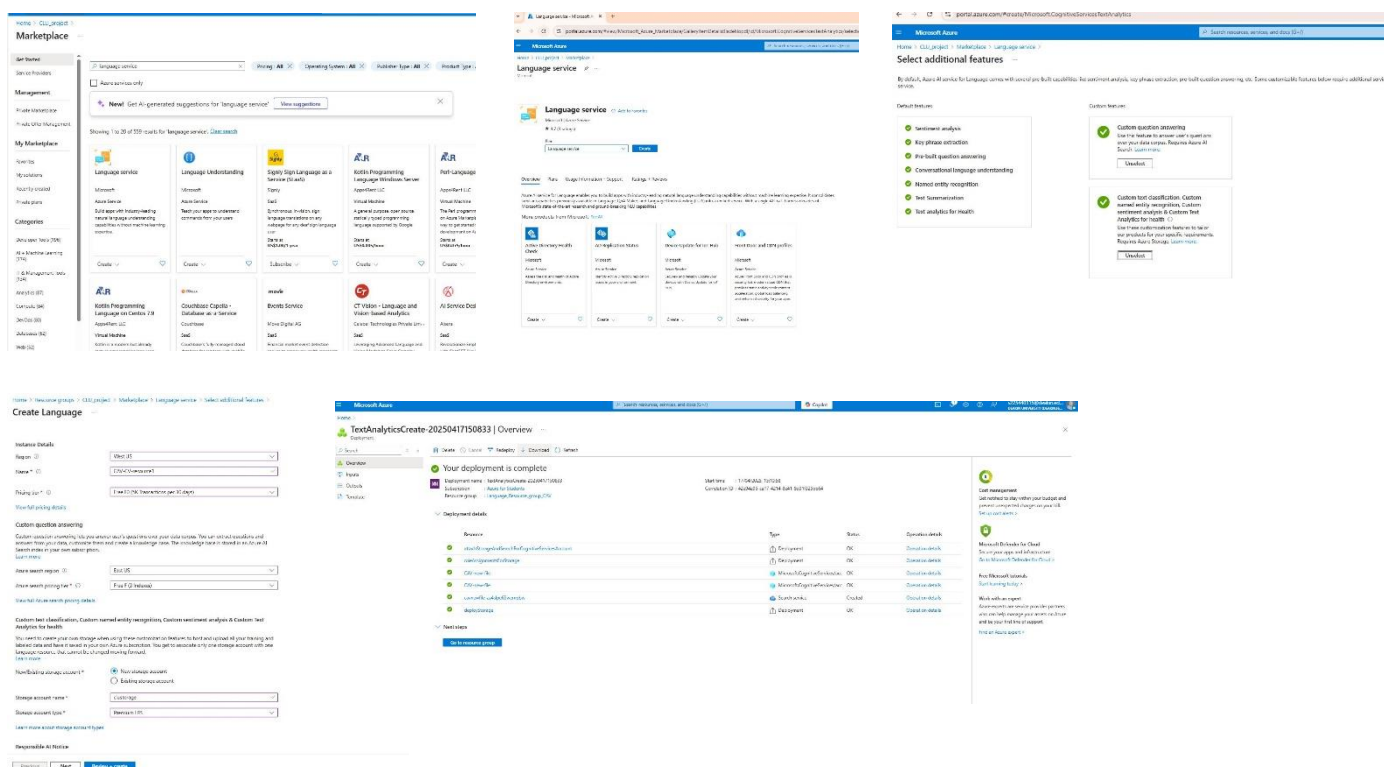
Create a language resource

Before you start using CLU, you will need an Azure AI Language resource.

Create a new resource from the Azure portal

1. Sign in to the **Azure portal** to create a new Azure AI Language resource.
2. resource group creation
3. Select **Create a new resource**
4. In the window that appears, search for **Language service**
5. Select **Create**
6. Create a language resource with following details.





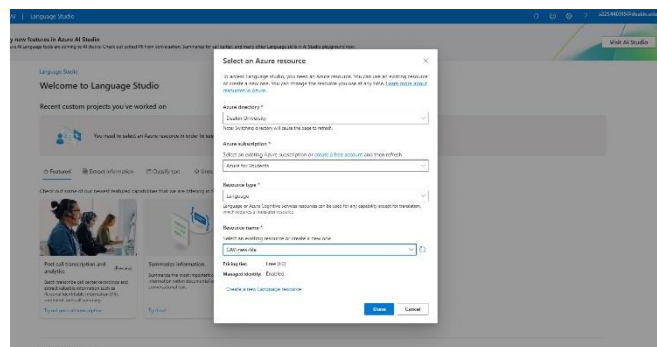
B. Create a new Language resource from Language Studio

If it's your first time logging in, you'll see a window in Language Studio that will let you choose an existing Language resource or create a new one. You can also create a resource by clicking the settings icon in the top-right corner, selecting Resources, then clicking Create a new resource.

Create a language resource with following details.

Sign in to Language Studio

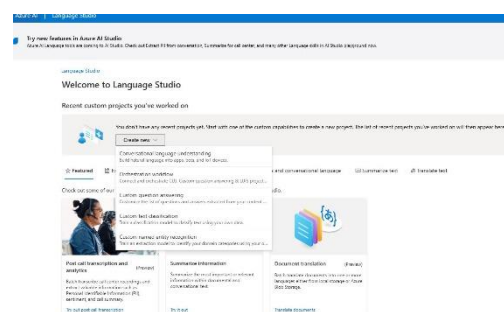
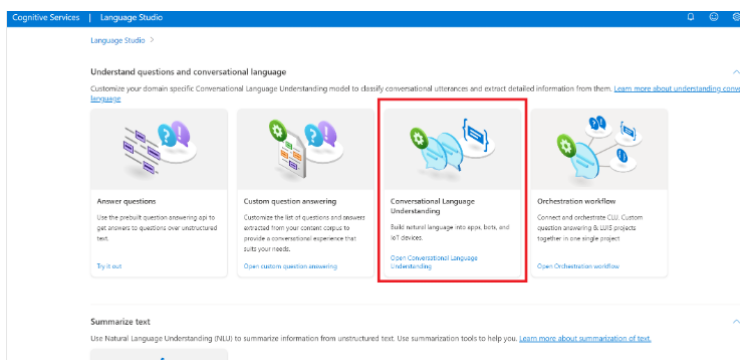
1. Go to the language studios and sign in with your Azure account.
2. In the Choose a language resource window that appears, find your Azure subscription, and choose your language resource. If you don't have a resource, you can create a new one.



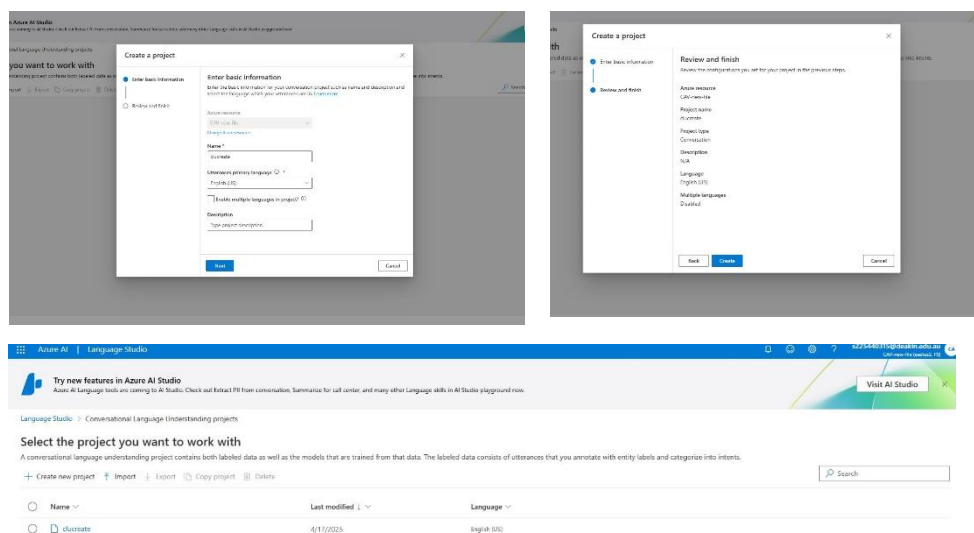
• Create a conversational language project

Once you have a language resource created, create a Conversational Language Understanding project.

A. In language Studio, find the section named Understand questions and conversational language and select Conversational language understanding.



A. This will bring you to the Conversational language understanding projects page. Select Create new project.



To create a new project, you need to provide the following details:

Value	Description
Name	A name for your project. Once it's created, it can't be changed
Description	Optional project description.
Utterances primary language	The primary language of your project. Your training data should primarily be in this language.
Enable multiple languages	Whether you would like to enable your project to support multiple languages at once.

Once you're done, select **Next** and review the details. Select **create** to complete the process. You should now see the **Build Schema** screen in your project.

• Import project

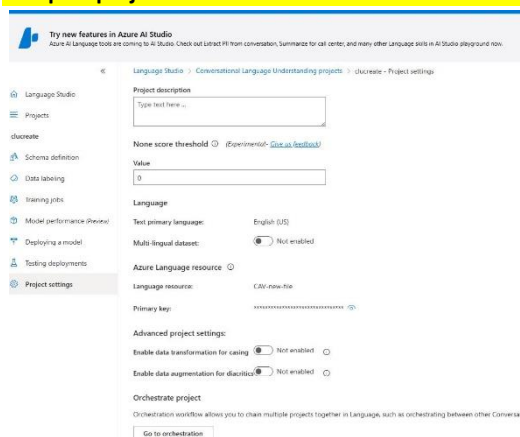
You can export a Conversational Language Understanding project as a JSON file at any time by going to the conversation projects page, selecting a project, and from the top menu, clicking on Export.

That project can be reimported as a new project. If you import a project with the exact same name, it replaces the project's data with the newly imported project's data.

If you have an existing LUIS application, you can *import* the LUIS application JSON to Conversational Language Understanding directly, and it will create a Conversation project with all the pieces that are currently available: Intents, ML entities, and utterances. See the LUIS migration articles for more information.

To import a project, select the arrow button next to Create a new project and select Import, then select the LUIS or Conversational Language Understanding JSON file.

• Export project



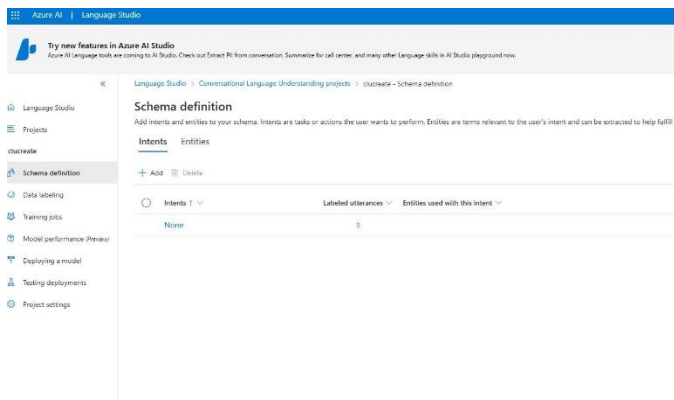
- Go to your project settings page in Language studio.
- You can see project details.
- In this page you can update project description and enable/disable Multi-lingual dataset in project settings.
- You can also retrieve your resource primary key from this page.

2.1.3 Explore how CLU processes text inputs and recognizes user intents and entities.

before CLU (Conversational Language Understanding) can process and make sense of input text, you first need to build schemas that guide how it understands different kinds of input.

• Build a Schema

In conversational language understanding projects, the *schema* is defined as the combination of intents and entities within your project. Schema design is a crucial part of your project's success. When creating a schema, you want think about which intents and entities should be included in your project.



A. Guidelines and recommendations

Consider the following guidelines when picking intents for your project:

1. Create distinct, separable intents. An intent is best described as action the user wants to perform. Think of the project you're building and identify all the different actions your users may take when interacting with your project. Sending, calling, and canceling are all actions that are best represented as different intents. "Cancelling an order" and "canceling an appointment" are very similar, with the distinction being *what* they are canceling. Those two actions should be represented under the same intent, *Cancel*.
2. Create entities to extract relevant pieces of information within your text. The entities should be used to capture the relevant information needed to fulfil your user's action. For example, *order* or *appointment* could be different things a user is trying to cancel, and you should create an entity to capture that piece of information.

You can *"send"* a message, *"send"* an email, or *"send"* a package. Creating an intent to capture each of those requirements will not scale over time, and you should use entities to identify *what* the user was sending. The combination of intents and entities should determine your conversation flow.

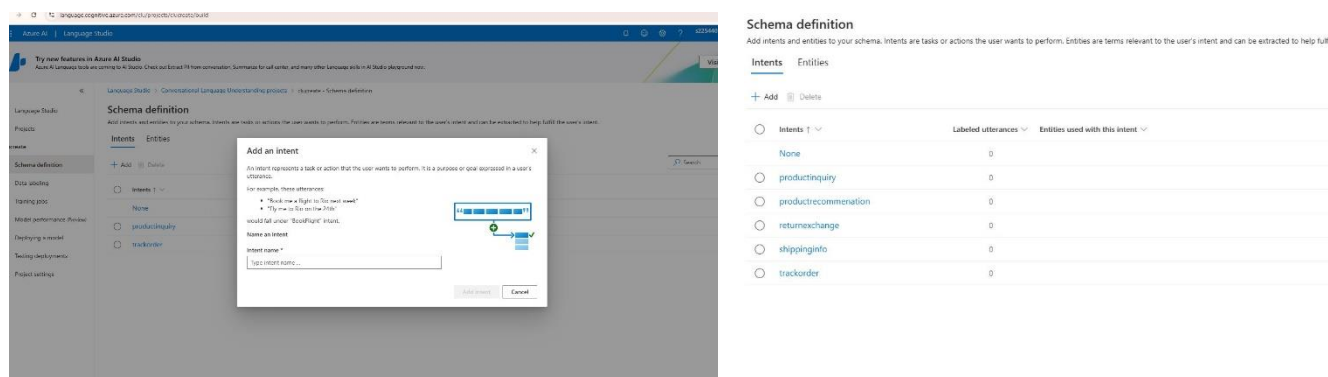
For example, consider a company where the bot developers have identified the three most common actions their users take when using a calendar:

- Setup new meetings
- Respond to meeting requests
- Cancel meetings

They might create an intent to represent each of these actions. They might also include entities to help complete these actions, such as:

- Meeting attendants
- Date
- Meeting durations

B. Add intents

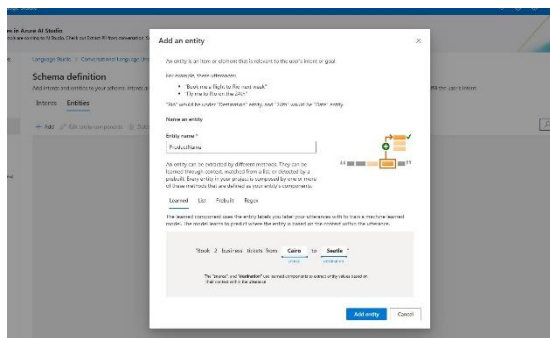


To build a project schema within Language Studio:

1. Select Schema definition from the left side menu.
2. From the top pivots, you can change the view to be Intents or Entities.
3. To create an intent, select Add from the top menu. You will be prompted to type in a name before completing creating the intent.
4. Repeat the above step to create all the intents to capture all the actions that you think the user will want to perform while using the project.

- When you select the intent, you will be directed to the data labelling page, with a filter set for the intent you selected. You can add examples for intents and label them with entities.
-

C. Add entities



- Move to **Entities** pivot from the top of the page.
- To add an entity, select **Add** from the top menu. You will be prompted to type in a name before completing creating the entity.
- After creating an entity, you'll be routed to the entity details page where you can define the composition settings for this entity.
- Every entity can be defined by multiple components: learned, list or prebuilt. A learned component is added to all your entities once you label them in your utterances.
- You can add a list or prebuilt component to each entity.

D. Add prebuilt component

To add a **prebuilt** component, select **Add new prebuilt** and from the drop-down menu, select the prebuilt type to you want to add to this entity.

E. Add list component

To add a list component, select **Add new list**. You can add multiple lists to each entity.

- To create a new list, in the *Enter value* text box enter this is the normalized value that will be returned when any of the synonyms values is extracted.

- From the *language* drop-down menu, select the language of the synonyms list and start typing in your synonyms and hit enter after each one. It is recommended to have synonyms lists in multiple languages.

F. Define entity options

- 3. Configure Entity Options:**
In the entity details page, switch to the *Entity Options* tab. Set how overlapping predictions are resolved and click **Save**.
- 4. Edit Entities Anytime:**
You can return later to modify or delete entity components from the top menu.

2.2 Exploring CLU Applications

• Select a real-world use case where CLU could be applied:

- Retail Chatbot – Assisting customers in finding products.
- Healthcare Assistant – Answering patient inquiries. – Travel Booking System
- Helping users with reservations.

• Universal Steps to Build CLU Use Cases Across Any Domain with MSA (Microsoft Azure)

• Utterance and label your data

• Creating Schemas

A schema in Conversational Language Understanding (CLU) defines the structure of interactions by combining intents (user goals) and entities (key information). Below is how schemas are designed for three domains:

1. Retail Chatbot – Assisting Customers in Finding Products

Objective: Understand shopping queries, extract product details, and provide recommendations.

Intents (User Goals)

Intent	Example Queries
FindProduct	"Show me wireless headphones", "I need a formal shirt"
CompareProducts	"Which phone has better battery: iPhone or Samsung?"
FilterProducts	"Show laptops under ₹50K with 16GB RAM"

Intent	Example Queries
GetRecommendations	"Suggest a gift for my mom's birthday"
TrackOrder	"Where is my order #12345?"

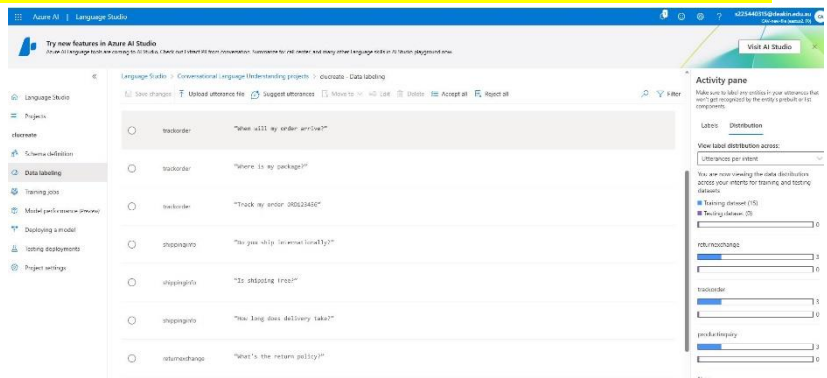
Entities (Key Details)

Entity	Purpose	Examples
ProductType	Category of item	"smartphone", "running shoes"
Brand	Manufacturer	"Nike", "Apple"
PriceRange	Budget constraints	"under ₹1000", "mid-range"
Features	Specific attributes	"waterproof", "wireless"
Color/Size	Variants	"blue", "size XL"

Example Schema Implementation:

● Label the utterances in Language Studio

- Once you have built schema for your project, you should add training utterances to your project. The utterances should be similar to what your users use when interacting with the project. When you add an utterance, you have to assign which intent it belongs to. After the utterance is added, label the words within your utterance that you want to extract as entities.
- Data labeling is a crucial step in development lifecycle; this data are used in the next step when training your model so that your model can learn from the labeled data. If you already have labeled utterances, you can directly import into your project, but you need to make sure that your data follows the accepted data format. See create project to learn more about importing labeled data into your project. Labeled data informs the model how to interpret text, and is used for training and evaluation.



● Data labelling guidelines

After building your schema and creating your project, you need to label your data. Labeling your data is important so your model knows which words and sentences are associated with the intents and entities in your project. Spend time labeling your utterances - introducing and refining the data that is used in training your models.

As you add utterances and label them, keep in mind:

- The machine learning models generalize based on the labeled examples you provide it; the more examples you provide, the more data points the model has to make better generalizations.
 - The precision, consistency and completeness of your labeled data are key factors to determining model performance.
- **Label precisely:** Label each intent and entity to its right type always. Only include what you want classified and extracted, avoid unnecessary data in your labels.
 - **Label consistently:** The same entity should have the same label across all the utterances.
 - **Label completely:** Provide varied utterances for every intent. Label all the instances of the entity in all your utterances.

● Clearly label utterances

- Ensure that the concepts that your entities refer to are well defined and separable. Check if you can easily determine the differences reliably. If you can't, this lack of distinction might indicate that the learned component will also have difficulty.
- If there's a similarity between entities, ensure that there's some aspect of your data that provides a signal for the difference between them.

For example, if you built a model to book flights, a user might use an utterance like "I want a flight from Boston to Seattle." The *origin city* and *destination city* for such utterances would be expected to be similar. A signal to differentiate *origin city* might be that the word *from* often precedes it.

- Ensure that you label all instances of each entity in both your training and testing data. One approach is to use the search function to find all instances of a word or phrase in your data to check if they're correctly labeled.

- Label test data for entities that have no learned component and also for the entities that do. This practice helps to ensure that your evaluation metrics are accurate.
- For multilingual project, adding utterances in other languages increases the model's performance in these languages, but avoid duplicating your data across all the languages you would like to support. For example, to improve a calendar bot's performance with users, a developer might add examples mostly in English, and a few in Spanish or French as well. They might add utterances such as:
 - "Set a meeting with **Matt** and **Kevin tomorrow** at **12 PM.**" (English)
 - "Reply as **tentative** to the **weekly update** meeting." (English)
 - "Cancelar mi **próxima** reunión." (Spanish)

• How to label your utterances

Use the following steps to label your utterances:

1. Go to your project page in language studio.
2. From the left side menu, select **Data labeling**. In this page, you can start adding your utterance and labeling them. You can also upload your utterance directly by clicking on **Upload utterance file** from the top menu, make sure it follows the accepted format
3. From the top pivots, you can change the view to be **training set** or **testing set**. Learn more about training and testing set and how they're used for model training and evaluation.

Option	Description
Label using a brush	Select the brush icon next to an entity in the right pane, then highlight the text in the utterance you want to label.
Label using inline menu	Highlight the word you want to label as an entity, and a menu appears. Select the entity you want to label these words with.

Note: If you are planning on using **Automatically split the testing set from training data** splitting, add all your utterances to the training set.

4. From the **Select intent** dropdown menu, select one of the intents, the language of the utterance (for multilingual projects), and the utterance itself. Press the enter key in the utterance's text box to add the utterance.

5. You have two options to label entities in an utterance:

1. In the right side pane, under the **Labels** pivot, you can find all the entity types in your project and the count of labelled instances per each.
2. Under the **Distribution** pivot you can view the distribution across training and testing sets. You have two options for viewing:
 - *Total instances per labeled entity* where you can view count of all labeled instances of a specific entity.
 - *Unique utterances per labeled entity* where each utterance is counted if it contains at least one labeled instance of this entity.
 - *Utterances per intent* where you can view count of utterances per intent.

Note: List and prebuilt components are not shown in the data labeling page, and all labels here only apply to the **learned component**.

To remove a label:

1. From within your utterance, select the entity you want to remove a label from.
2. Scroll through the menu that appears, and select **Remove label**.

To delete an entity:

1. Select the entity you want to edit in the right side pane.
2. Select the three dots next to the entity, and select the option you want from the drop-down menu.

• Understanding and Labeling Retail Queries for CLU Models

A. Simple Query (Single Intent + Entities)

Utterance:

"Show me waterproof backpacks under ₹2,000"

Labels:

- **Intent:** FindProduct
- **Entities:**
 - ProductType: "backpacks"
 - Features: "waterproof"
 - PriceRange: "under ₹2,000"

B. Complex Query (Multiple Entities)

Utterance:

"Compare iPhone 15 Pro and Samsung Galaxy S23 Ultra cameras"

Labels:

- **Intent:** CompareProducts
- **Entities:**
 - Brand: "iPhone", "Samsung Galaxy"
 - Model: "15 Pro", "S23 Ultra"
 - Features: "cameras"

C. Ambiguous Query (Requires Clarification)

Utterance:

"I want a small bag"

Possible Labels:

- **Intent:** FindProduct
- **Entities:**
 - ProductType: "bag"
 - Size: "small" (*but is "small" a size or style? Clarify!*)

4. Tools for Labeling Retail Utterances

1. **Manual Labeling Tools:**
 - **Label Studio** (Open-source)
 - **Azure CLU Studio** (Pre-built retail schema support)
 - **Amazon SageMaker Ground Truth**

• Suggest utterances with Azure OpenAI

In CLU, use Azure OpenAI to suggest utterances to add to your project using GPT models. You first need to get access and create a resource in Azure OpenAI. You'll then need to create a deployment for the GPT models. Follow the pre-requisite steps [here](#).

Before you get started, the suggest utterances feature is only available if your Language resource is in the following regions:

- East US
- South Central US
- West Europe

In the Data Labeling page:

1. Select the **Suggest utterances** button. A pane opens up on the right side prompting you to select your Azure OpenAI resource and deployment.
2. On selection of an Azure OpenAI resource, select **Connect**, which allows your language resource to have direct access to your Azure OpenAI resource. It assigns your language resource the role of Cognitive Services User to your Azure OpenAI resource, which allows your current Language resource to have access to Azure OpenAI's service. If the connection fails, follow these [steps](#) below to add the right role to your Azure OpenAI resource manually.
3. Once the resource is connected, select the deployment. The recommended model for the Azure OpenAI deployment is gpt-35-turbo-instruct.
4. Select the intent you'd like to get suggestions for. Make sure the intent you have selected has at least 5 saved utterances to be enabled for utterance suggestions. The suggestions provided by Azure OpenAI are based on the **most recent utterances** you've added for that intent.
5. Select **Generate utterances**. Once complete, the suggested utterances show up with a dotted line around it, with the note *Generated by AI*. Those suggestions need to be accepted or rejected. Accepting a suggestion simply adds it to your project, as if you had added it yourself. Rejecting it deletes the suggestion entirely. Only accepted utterances are part of your project and used for training or testing. You can accept or reject by clicking on the green check or red cancel buttons beside each utterance. You can also use the Accept all and Reject all buttons in the toolbar.

Using this feature entails a charge to your Azure OpenAI resource for a similar number of tokens to the suggested utterances generated. Details for Azure OpenAI's pricing can be found [here](#).

• Add required configurations to Azure OpenAI resource

Azure portal Language Studio

Your Language resource must have identity management, to enable it using the [Azure portal](#):

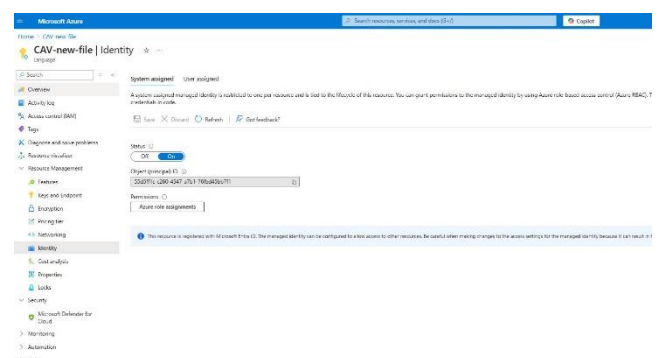
1. Go to your Language resource
2. From left hand menu, under **Resource Management** section, select **Identity**
3. From **System assigned** tab, make sure to set **Status** to On

Azure portal Language Studio

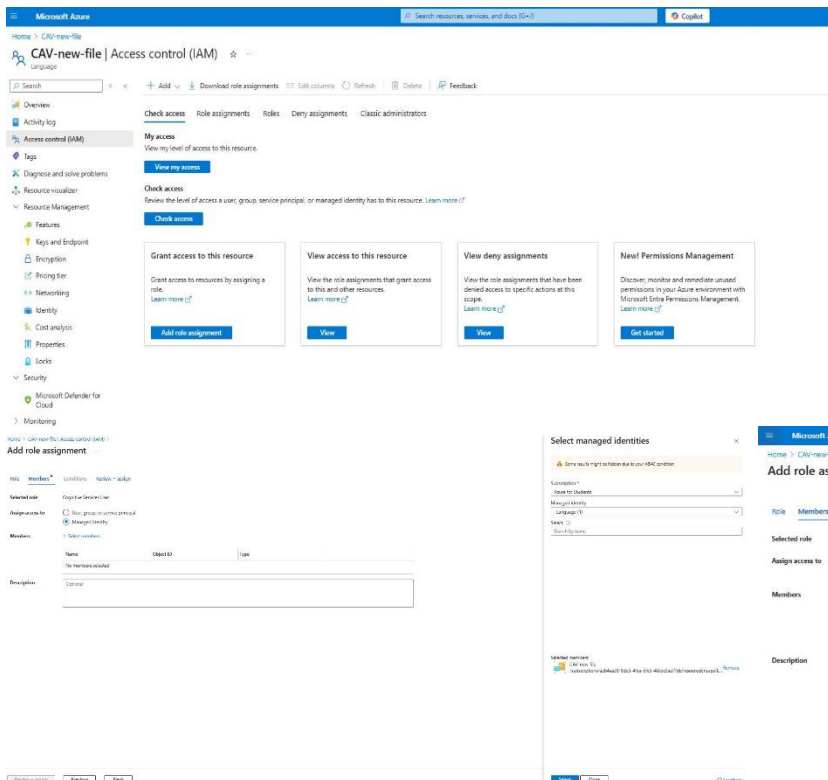
Your Language resource must have identity management, to enable it using [Language Studio](#):

1. Select the settings icon in the top right corner of the screen
2. Select **Resources**
3. Select the check box **Managed Identity** for your Language resource.

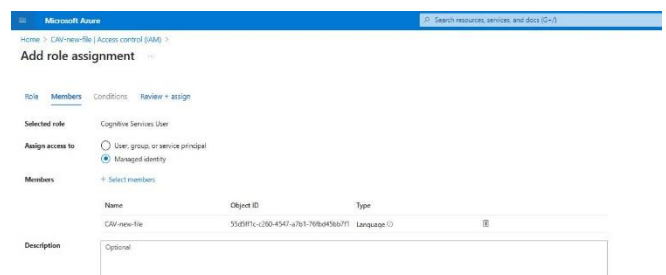
- If connecting your Language resource to an Azure OpenAI resource fails, follow these steps:
- Enable identity management for your Language resource using the following options:



After enabling managed identity, assign the role Cognitive Services User to your Azure OpenAI resource using the managed identity of your language resource.



1. Sign in to the Azure portal and navigate to your Azure OpenAI resource.
2. Select the Access Control (IAM) tab on the left.
3. Select Add > Add role assignment.
4. Select "Job function roles" and click Next.
5. Select Cognitive Services User from the list of roles and click Next.
6. Select Assign access to "Managed identity" and select "Select members".
7. Under "Managed identity" select "Language".
8. Search for your resource and select it. Then select the Select button below and next to complete the process.
9. Review the details and select Review + Assign.



Microsoft Azure																																																																																																								
Home > CAV-new-file > Access control (IAM)																																																																																																								
Add role assignment																																																																																																								
Role Members Conditions Review + assign A role definition is a collection of permissions. You can use the built-in roles or you can create your own custom roles. Learn more Job function roles Privileged administrator roles Grant access to Azure resources based on job function, such as the ability to create virtual machines. Search by role name, description, permissions, or ID <table> <thead> <tr> <th>Name</th><th>Description</th><th>Type</th><th>Category</th><th>Details</th></tr> </thead> <tbody> <tr> <td>Reader</td><td>View all resources, but does not allow you to make any changes.</td><td>Built-in role</td><td>General</td><td>View</td></tr> <tr> <td>App Compliance Automation Administrator</td><td>Create, read, download, modify and delete reports objects and related other resource objects.</td><td>Built-in role</td><td>None</td><td>View</td></tr> <tr> <td>App Compliance Automation Reader</td><td>Read, download the reports objects and related other resource objects.</td><td>Built-in role</td><td>None</td><td>View</td></tr> <tr> <td>Azure AI Administrator</td><td>A Built-in Role that has all control plane permissions to work with Azure AI and its dependencies.</td><td>Built-in role</td><td>None</td><td>View</td></tr> <tr> <td>Azure AI Developer</td><td>Can perform all actions within an Azure AI resource besides managing the resource itself.</td><td>Built-in role</td><td>None</td><td>View</td></tr> <tr> <td>Azure AI Enterprise Network Connection Approver</td><td>Can approve private endpoint connections to Azure AI common dependency resources.</td><td>Built-in role</td><td>None</td><td>View</td></tr> <tr> <td>Cognitive Services Contributor</td><td>Lets you create, read, update, delete and manage keys of Cognitive Services.</td><td>Built-in role</td><td>AI - Machine Learning</td><td>View</td></tr> <tr> <td>Cognitive Services Custom Vision Contributor</td><td>Full access to the project, including the ability to view, create, edit, or delete projects.</td><td>Built-in role</td><td>AI - Machine Learning</td><td>View</td></tr> <tr> <td>Cognitive Services Custom Vision Deployment</td><td>Publish, update or report models. Deployment can view the project but can't update.</td><td>Built-in role</td><td>AI - Machine Learning</td><td>View</td></tr> <tr> <td>Cognitive Services Custom Vision Labeler</td><td>View, edit training images and create, add, remove, or delete the image tags. Labelers can view the project but can't update anything other than training images and tags.</td><td>Built-in role</td><td>AI - Machine Learning</td><td>View</td></tr> <tr> <td>Cognitive Services Custom Vision Reader</td><td>Read-only access to the project. Readers can't create or update the project.</td><td>Built-in role</td><td>AI - Machine Learning</td><td>View</td></tr> <tr> <td>Cognitive Services Custom Vision Trainer</td><td>View, edit projects and train the models, including the ability to publish, update, report the models. Trainers can't create or delete the project.</td><td>Built-in role</td><td>AI - Machine Learning</td><td>View</td></tr> <tr> <td>Cognitive Services Data Reader (Preview)</td><td>Allows to read data plane APIs, but not any control plane APIs for Microsoft Cognitive Services. This role is in preview and subject to change.</td><td>Built-in role</td><td>None</td><td>View</td></tr> <tr> <td>Cognitive Services Data Reader</td><td>Lets you read Cognitive Services data.</td><td>Built-in role</td><td>Previous</td><td>View</td></tr> <tr> <td>Cognitive Services Face Contributor</td><td>Full access to perform all Face APIs.</td><td>Built-in role</td><td>None</td><td>View</td></tr> <tr> <td>Cognitive Services Face Recognizer</td><td>Lets you perform detect, verify, identify, group, and find similar operations on Face API. This role does not allow create or delete operations, which makes it well suited for end.</td><td>Built-in role</td><td>AI - Machine Learning</td><td>View</td></tr> <tr> <td>Cognitive Services Immersive Reader User</td><td>Provides access to create Immersive Reader sessions and call APIs.</td><td>Built-in role</td><td>None</td><td>View</td></tr> <tr> <td>Cognitive Services Language Overview</td><td>Has access to all Read, Text, Write, Display and Delete functions under Language portal.</td><td>Built-in role</td><td>None</td><td>View</td></tr> <tr> <td>Creation Services Language Reader</td><td>Has access to Read and Text functions under Language portal.</td><td>Built-in role</td><td>None</td><td>View</td></tr> </tbody> </table>					Name	Description	Type	Category	Details	Reader	View all resources, but does not allow you to make any changes.	Built-in role	General	View	App Compliance Automation Administrator	Create, read, download, modify and delete reports objects and related other resource objects.	Built-in role	None	View	App Compliance Automation Reader	Read, download the reports objects and related other resource objects.	Built-in role	None	View	Azure AI Administrator	A Built-in Role that has all control plane permissions to work with Azure AI and its dependencies.	Built-in role	None	View	Azure AI Developer	Can perform all actions within an Azure AI resource besides managing the resource itself.	Built-in role	None	View	Azure AI Enterprise Network Connection Approver	Can approve private endpoint connections to Azure AI common dependency resources.	Built-in role	None	View	Cognitive Services Contributor	Lets you create, read, update, delete and manage keys of Cognitive Services.	Built-in role	AI - Machine Learning	View	Cognitive Services Custom Vision Contributor	Full access to the project, including the ability to view, create, edit, or delete projects.	Built-in role	AI - Machine Learning	View	Cognitive Services Custom Vision Deployment	Publish, update or report models. Deployment can view the project but can't update.	Built-in role	AI - Machine Learning	View	Cognitive Services Custom Vision Labeler	View, edit training images and create, add, remove, or delete the image tags. Labelers can view the project but can't update anything other than training images and tags.	Built-in role	AI - Machine Learning	View	Cognitive Services Custom Vision Reader	Read-only access to the project. Readers can't create or update the project.	Built-in role	AI - Machine Learning	View	Cognitive Services Custom Vision Trainer	View, edit projects and train the models, including the ability to publish, update, report the models. Trainers can't create or delete the project.	Built-in role	AI - Machine Learning	View	Cognitive Services Data Reader (Preview)	Allows to read data plane APIs, but not any control plane APIs for Microsoft Cognitive Services. This role is in preview and subject to change.	Built-in role	None	View	Cognitive Services Data Reader	Lets you read Cognitive Services data.	Built-in role	Previous	View	Cognitive Services Face Contributor	Full access to perform all Face APIs.	Built-in role	None	View	Cognitive Services Face Recognizer	Lets you perform detect, verify, identify, group, and find similar operations on Face API. This role does not allow create or delete operations, which makes it well suited for end.	Built-in role	AI - Machine Learning	View	Cognitive Services Immersive Reader User	Provides access to create Immersive Reader sessions and call APIs.	Built-in role	None	View	Cognitive Services Language Overview	Has access to all Read, Text, Write, Display and Delete functions under Language portal.	Built-in role	None	View	Creation Services Language Reader	Has access to Read and Text functions under Language portal.	Built-in role	None	View
Name	Description	Type	Category	Details																																																																																																				
Reader	View all resources, but does not allow you to make any changes.	Built-in role	General	View																																																																																																				
App Compliance Automation Administrator	Create, read, download, modify and delete reports objects and related other resource objects.	Built-in role	None	View																																																																																																				
App Compliance Automation Reader	Read, download the reports objects and related other resource objects.	Built-in role	None	View																																																																																																				
Azure AI Administrator	A Built-in Role that has all control plane permissions to work with Azure AI and its dependencies.	Built-in role	None	View																																																																																																				
Azure AI Developer	Can perform all actions within an Azure AI resource besides managing the resource itself.	Built-in role	None	View																																																																																																				
Azure AI Enterprise Network Connection Approver	Can approve private endpoint connections to Azure AI common dependency resources.	Built-in role	None	View																																																																																																				
Cognitive Services Contributor	Lets you create, read, update, delete and manage keys of Cognitive Services.	Built-in role	AI - Machine Learning	View																																																																																																				
Cognitive Services Custom Vision Contributor	Full access to the project, including the ability to view, create, edit, or delete projects.	Built-in role	AI - Machine Learning	View																																																																																																				
Cognitive Services Custom Vision Deployment	Publish, update or report models. Deployment can view the project but can't update.	Built-in role	AI - Machine Learning	View																																																																																																				
Cognitive Services Custom Vision Labeler	View, edit training images and create, add, remove, or delete the image tags. Labelers can view the project but can't update anything other than training images and tags.	Built-in role	AI - Machine Learning	View																																																																																																				
Cognitive Services Custom Vision Reader	Read-only access to the project. Readers can't create or update the project.	Built-in role	AI - Machine Learning	View																																																																																																				
Cognitive Services Custom Vision Trainer	View, edit projects and train the models, including the ability to publish, update, report the models. Trainers can't create or delete the project.	Built-in role	AI - Machine Learning	View																																																																																																				
Cognitive Services Data Reader (Preview)	Allows to read data plane APIs, but not any control plane APIs for Microsoft Cognitive Services. This role is in preview and subject to change.	Built-in role	None	View																																																																																																				
Cognitive Services Data Reader	Lets you read Cognitive Services data.	Built-in role	Previous	View																																																																																																				
Cognitive Services Face Contributor	Full access to perform all Face APIs.	Built-in role	None	View																																																																																																				
Cognitive Services Face Recognizer	Lets you perform detect, verify, identify, group, and find similar operations on Face API. This role does not allow create or delete operations, which makes it well suited for end.	Built-in role	AI - Machine Learning	View																																																																																																				
Cognitive Services Immersive Reader User	Provides access to create Immersive Reader sessions and call APIs.	Built-in role	None	View																																																																																																				
Cognitive Services Language Overview	Has access to all Read, Text, Write, Display and Delete functions under Language portal.	Built-in role	None	View																																																																																																				
Creation Services Language Reader	Has access to Read and Text functions under Language portal.	Built-in role	None	View																																																																																																				

• Train your conversational language understanding model

- After you have completed labeling your utterances, you can start training a model. Training is the process where the model learns from your labeled utterances.
- To train a model, start a training job. Only successfully completed jobs create a model. Training jobs expire after seven days, after this time you will no longer be able to retrieve the job details. If your training job completed successfully and a model was created, it won't be affected by the job expiring. You can only have one training job running at a time, and you can't start other jobs in the same project.
- The training times can be anywhere from a few seconds when dealing with simple projects, up to a couple of hours when you reach the maximum limit of utterances.
- Model evaluation is triggered automatically after training is completed successfully. The evaluation process starts by using the trained model to run predictions on the utterances in the testing set, and compares the predicted results with the provided labels (which establishes a baseline of truth).

• Balance training data

When it comes to training data, try to keep your schema well balanced. Including large quantities of one intent and very few of another results in a model that's biased toward particular intents.

To address this scenario, you might need to down sample your training set. Or you might need to add to it. To down sample, you can:

- Get rid of a certain percentage of the training data randomly.
- Analyze the dataset and remove overrepresented duplicate entries, which is a more systematic manner. To add to the training set, in Language Studio, on the **Data labeling** tab, select **Suggest utterances**. Conversational Language Understanding sends a call to Azure Open AI to generate similar utterances.

You should also look for unintended "patterns" in the training set. For example, look to see if the training set for a particular intent is all lowercase or starts with a particular phrase. In such cases, the model you train might learn these unintended biases in the training set instead of being able to generalize.

We recommend that you introduce casing and punctuation diversity in the training set. If your model is expected to handle variations, be sure to have a training set that also reflects that diversity. For example, include some utterances in proper casing and some in all lowercase.

• Data splitting

Before you start the training process, labeled utterances in your project are divided into a training set and a testing set. Each one of them serves a different function. The **training set** is used in training the model, this is the set from which the model learns the labeled utterances. The **testing set** is a blind set that isn't introduced to the model during training but only during evaluation.

After the model is trained successfully, the model can be used to make predictions from the utterances in the testing set. These predictions are used to calculate evaluation metrics. It is recommended to make sure that all your intents and entities are adequately represented in both the training and testing set.

Conversational language understanding supports two methods for data splitting:

- **Automatically splitting the testing set from training data:** The system will split your tagged data between the training and testing sets, according to the percentages you choose. The recommended percentage split is 80% for training and 20% for testing.
- **Note**
- If you choose the **Automatically splitting the testing set from training data** option, only the data assigned to training set will be split according to the percentages provided.
- **Use a manual split of training and testing data:** This method enables users to define which utterances should belong to which set. This step is only enabled if you have added utterances to your testing set during labelling.

• Training models

CLU supports two modes for training your models

- **Standard training:** uses fast machine learning algorithms to train your models relatively quickly. This is currently only available for **English** and is disabled for any project that doesn't use English (US), or English (UK) as its primary language. This training option is free of charge. Standard training allows you to add utterances and test them quickly at no cost. The evaluation scores shown should guide you on where to make changes in your project and add more utterances. Once you've iterated a few times and made incremental improvements, you can consider using advanced training to train another version of your model.
- **Advanced training:** uses the latest in machine learning technology to customize models with your data. This is expected to show better performance scores for your models and will enable you to use the multi language capabilities of CLU as well. Advanced training is priced differently. See the pricing information for details.

Use the evaluation scores to guide your decisions. There might be times where a specific example is predicted incorrectly in advanced training as opposed to when you used standard training mode. However, if the overall evaluation results are better using a advanced, then it is recommended to use your final model. If that isn't the case and you are not looking to use any multilingual capabilities, you can continue to use model trained with standard mode.

Note

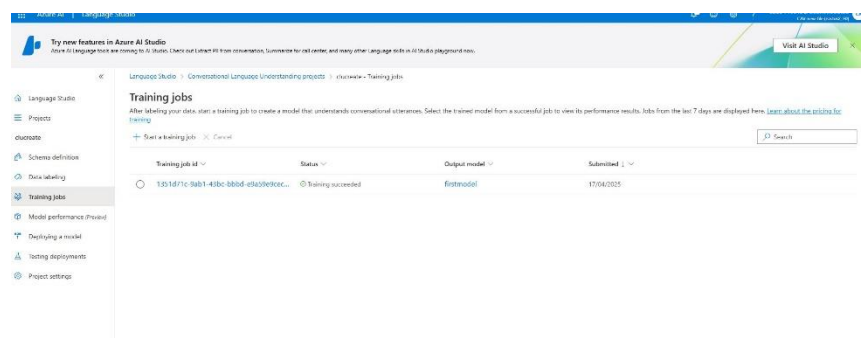
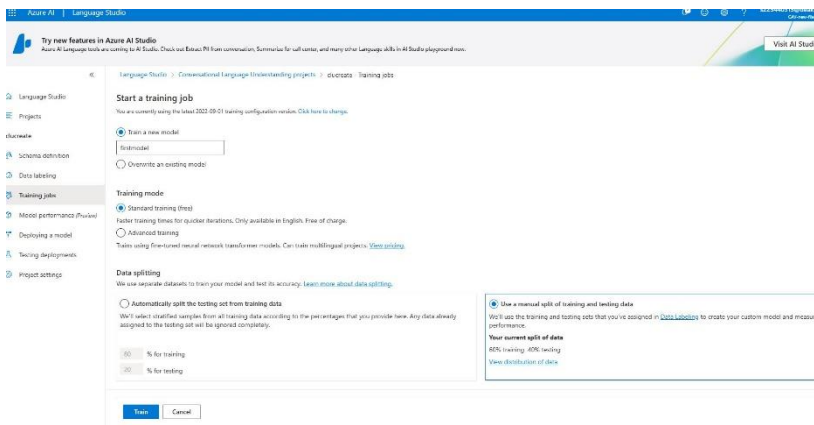
we should expect to see a difference in behaviours in intent confidence scores between the training modes as each algorithm calibrates their scores differently.

Train model

To start training your model from within the language Studios:

1. Select **Train model** from the left side menu.
2. Select **Start a training job** from the top menu.
3. Select **Train a new model** and enter a new model name in the text box. Otherwise, to replace an existing model with a model trained on the new data, select **Overwrite an existing model** and then select an existing model. Overwriting a trained model is irreversible, but it won't affect your deployed models until you deploy the new model.

4. Select training mode. You can choose **Standard training** for faster training, but it is only available for English. Or you can choose **Advanced training** which is supported for other languages and multilingual projects, but it involves longer training times. Learn more about training models.
5. Select a data splitting method. You can choose **Automatically splitting the testing set from training data** where the system will split your utterances between the training and testing sets, according to the specified percentages. Or you can **Use a manual split of training and testing data**, this option is only enabled if you have added utterances to your testing set when you labelled your utterances
6. Select the **Train** button.



Select the training job ID from the list. A panel will appear where you can check the training progress, job status, and other details for this job.

Note

- Only successfully completed training jobs will generate models.
- Training can take some time between a couple of minutes and couple of hours based on the count of utterances.
- You can only have one training job running at a time. You can't start other training jobs within the same project until the running job is completed.
- The machine learning used to train models is regularly updated. To train on a previous [configuration version](#), select **Select here to change** from the **Start a training job** page and choose a previous version.

Cancel training job

To cancel a training job from within Language Studio

1. In the **Train model** page, select the training job you want to cancel and select **Cancel** from the top menu.

View conversational language understanding model details

After model training is completed, you can view your model details and see how well it performs against the test set.

Note

Using the **Automatically split the testing set from training data** option may result in different model evaluation result every time you train a new model, as the test set is selected randomly from your utterances. To make sure that the evaluation is calculated on the same test set every time you train a model, make sure to use the **Use a manual split of training and testing data** option when starting a training job and define your **Testing set** when add your utterances.

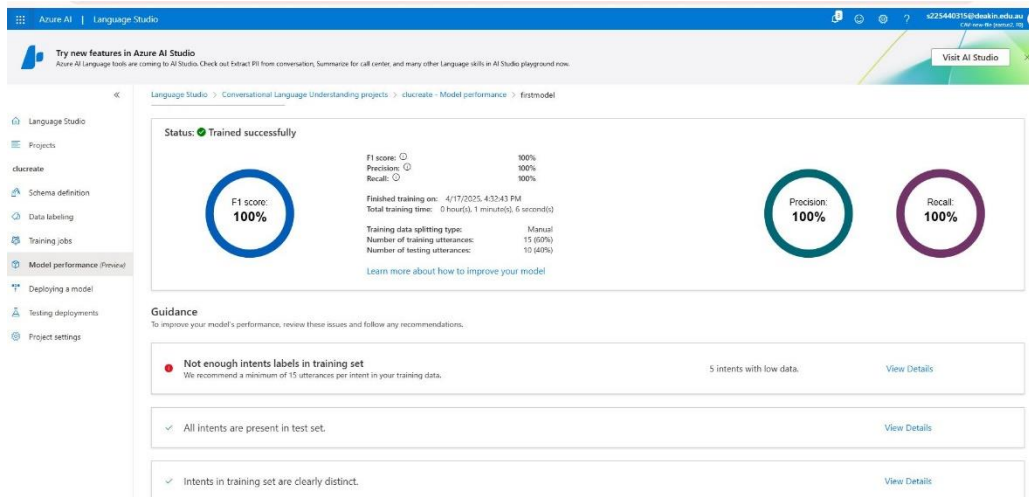
Model details

Language studio

1. Go to your project page in Language Studio
2. Select **Model performance** from the menu on the left side of the screen.

3. In this page you can only view the successfully trained models, F1 score of each model and [model expiration date](#). You can select the model name for more details about its performance. Models only include evaluation details if there was test data selected while training the model.

In this tab you can view the model's details such as: F1 score, precision, recall, date and time for the training job, total training time and number of training and testing utterances included in this training job. You can view details between intents or entities by selecting



Model Type at the top.

- You will also see guidance on how to improve the model. When clicking on view details a side panel will open to give more guidance on how to improve the model.

Load or export model data (Language Studio)

To **load** your model data:

- Select any model in the **model evaluation** page.
- Select the **Load model data** button.
- Confirm that you do not have any unsaved changes you need to capture in window that appears, and select **Load data**.
- Wait until your model data finishes loading back into your project. On completion, you'll be redirected back to the **Schema design** page.

To **export** your model data:

- Select any model in the **model evaluation** page.
- Select the **Export model data** button. Wait for the JSON snapshot of your model to be downloaded locally.

Delete model

- Language studio

To delete your model from within the Language Studio:

- Select **Model performance** from the left side menu.
- Select the **model name** you want to delete and select **Delete** from the top menu.
- In the window that appears, select **OK** to delete the model.

Deploy a model

Once you are satisfied with how your model performs, it's ready to be deployed, and query it for predictions from utterances. Deploying a model makes it available for use through the prediction API.

Deploy model

After you check the model's performance and decide it's ready to use, you need to **assign it to a deployment**. This lets you use the model through the prediction API.

It's a good idea to:

- Create a **deployment named production** for your best model — the one you want your system to use.
- Create a **deployment named staging** for models you're still testing.

You can have up to **10 deployments** in your project.

Unclear distinction between Intents in training set.

When the training data is similar for multiple intents, it can lead to lower accuracy because the intents may be frequently misclassified as each other.

Check the [confusion matrix](#) for details.

Recommendation

Review the following intents and consider merging them if they're similar. Otherwise, add more examples to better distinguish them from each other.

Intents with an unclear distinction in training set.

["QueryLastText" may be similar to "SearchMessages"](#)

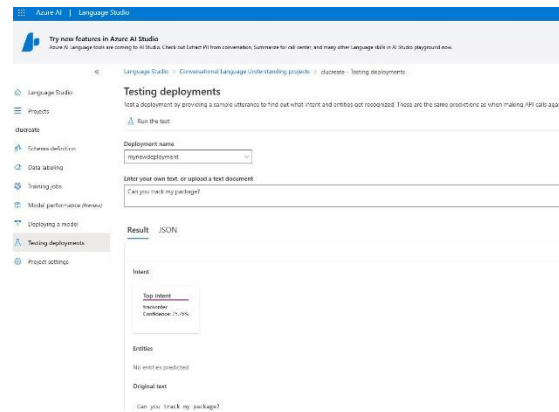
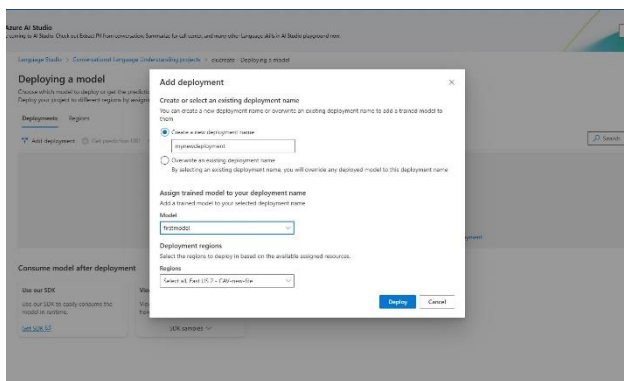
To deploy your model from within the language Studio:

1. Select **Deploying a model** from the left side menu.
2. Select **Add deployment** to start the **Add deployment** wizard.

To create a new deployment, choose "**Create a new deployment name**" and pick a trained model from the dropdown below. If you want to update an existing deployment with a new model, select "**Overwrite an existing deployment name**" to replace the current one.

Note:

Overwriting an existing deployment doesn't require changes to your prediction API call but the results you get will be based on the newly assigned model.

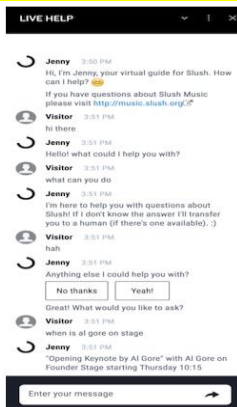


4. Select a trained model from the **Model** dropdown.
5. Select **Deploy** to start the deployment job.
6. After deployment is successful, an expiration date will appear next to it. Deployment expiration is when your deployed model will be unavailable to be used for prediction, which typically happens **twelve** months after a training configuration expires.

• Chatbot use cases in Retail Chatbot – Assisting customers in finding products.

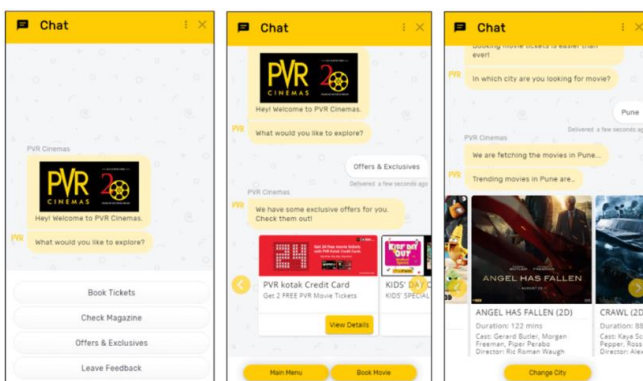
How can chatbots benefit your business list of amazing chatbot use cases from different industries. We'll show you how companies of all shapes and sizes are using chatbots for various tasks in the fields of Customer Service, Marketing and Sales.

1. Chatbots answer questions and inquiries



When a company hosts a large event, customer support becomes crucial—especially as audience size grows into the thousands. While guiding 20–50 attendees is manageable, handling queries from 5,000 or even 10,000 can overwhelm any support team. That's where a chatbot can make a real difference. With 24/7 availability and instant responses, it can handle common questions about tickets, venue details, speaker info, schedules, and nearby hotels. Slush, a global entrepreneurial event, used the LeadDesk chatbot to great success. In 2018, it managed 64% of all support queries and helped increase engagement by 55% compared to the previous year, efficiently serving around 20,000 attendees. This proves that no matter the scale, a chatbot can greatly ease the load on support teams while enhancing the customer experience.

2. Chatbots for Event Ticket Booking



Chatbots are transforming how customers book tickets for movies, concerts, sports event and shows. By handling repetitive queries like "What's playing today?", "Which seats are available?" and "What's the refund policy?", these AI assistants free up human agents for more complex tasks. Leading cinema chains like PVR in India have successfully implemented chatbots that not only answer questions but also complete ticket purchases, show trailers, and process refunds - creating a seamless customer experience. Beyond movies, this technology applies equally well to concerts, theater performances, sports matches and special events. The chatbot serves as a 24/7 virtual box office that never sleeps, automatically handling peak demand periods like new movie releases or playoff ticket sales. By integrating with seating charts and payment systems, these bots can show real-time availability, recommend best seats, and complete secure transactions. Additional features like trailer playback, special offer notifications, and personalized recommendations further enhance the user experience while

driving more sales. For event organizers, the benefits include reduced staffing costs, increased ticket sales, and valuable customer insights from interaction data. As consumers grow more comfortable with conversational commerce, ticket booking chatbots are becoming an essential tool for any venue or event management company looking to improve service while boosting revenue.

3. Chatbots for Product Discovery, Inventory Checks, and Personalized Recommendations

In today's e-commerce landscape, chatbots are revolutionizing the shopping experience by helping customers easily find products, check real-time inventory, and receive tailored recommendations. Many potential sales are lost simply because shoppers struggle to locate items or confirm availability, prompting them to abandon carts or overload customer support with basic queries.

Chatbots solve these challenges with instant, automated assistance. When a customer asks, *"Is this dress available in size medium?"* or *"What laptops do you recommend for graphic design?"*, the chatbot can:

- **Search inventory databases** in seconds to verify stock levels
- **Suggest similar or complementary items** based on purchase history or preferences
- **Guide shoppers** through filters like price, colour, or ratings

For businesses, this means **reduced pressure on human agents**, who can focus on complex issues while chatbots handle routine tasks. E-commerce sectors like **fashion, electronics, and grocery** benefit the most, as customers frequently seek size/colour availability or pairing suggestions before purchasing.

By integrating with **CRM and inventory systems**, chatbots ensure accurate, up-to-date responses, improving customer trust and satisfaction. Features like **personalized product carousels, instant stock alerts, and AI-driven "complete the look" recommendations** further boost engagement and conversions.

Ultimately, these AI assistants create a **frictionless shopping journey**—helping customers find what they need while driving sales and operational efficiency.

Key Benefits:

24/7 instant answers on inventory and product details

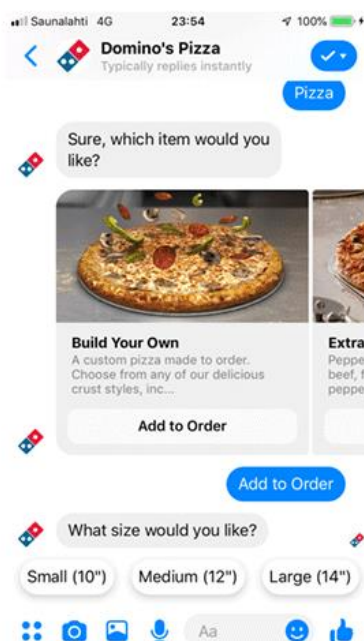
Higher conversions with smart, contextual recommendations

Agent workload reduced by ~40% (handling repetitive queries)

Seamless integration with e-commerce platforms (Shopify, WooCommerce, etc.)

Best For: Retailers, electronics stores, fashion brands, and grocery delivery services.

4. Chatbots for Seamless Customer Experiences



Modern chatbots are revolutionizing customer interactions by blending efficiency with engagement. Rather than purely functional transactions, today's bots guide users through delightful journeys - whether helping choose products via interactive quizzes, lightening the mood with humour and memes, or simplifying decisions with rich visual aids like product images and clickable options.

A prime example is Domino's Pizza, which transformed ordering into a frictionless experience through its Messenger chatbot. Customers enjoy:

One-click ordering without downloading apps

Conversational interface in familiar platforms

Live features like order tracking and store locator maps

This approach demonstrates how chatbots:

Remove friction from key processes

Delight customers through natural, enjoyable interactions

Build brand loyalty by meeting users where they already are

The result? Domino's saw significant increases in order frequency and customer satisfaction - proving that well-designed chatbots create both business value and memorable experiences.

Key Takeaways:

- Chatbots excel at making complex processes simple and enjoyable
- Integration with popular platforms (Messenger, WhatsApp) boosts accessibility
- The Domino's case study shows measurable impact on sales and satisfaction
- Balance of utility and personality drives engagement

Best For: Food delivery, retail, and service businesses looking to streamline customer journeys while strengthening brand connection

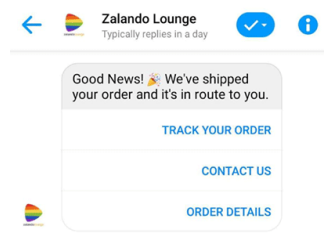
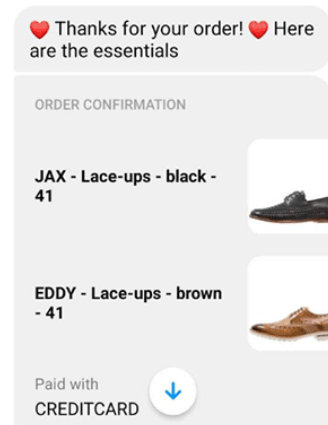
5. Chatbots can process return and exchange requests

One of the most common requests customer support agents get from customers is for refunds and exchanges. Companies often have a clear policy in place for processing such requests. This means, for customer support agents, performing most refunds and exchanges is a repetitive and monotonous task.

As such, a chatbot, if designed to carry out the company's refund and exchange policy, can certainly execute these tasks instead of customer support agents in your company.

In fact, you can 'bake' this function right into the chatbot's chat window with an option clearly labeled 'get a free refund'. And of course, in the same way, a chatbot can make a refund, it can also process item exchanges as well.

6. Chatbots can confirm orders and track shipping



After buying a product, customers may want to know how much time remains until their product ships.

Usually, there is a complicated process in place to do that. Customers have to go through their email to find the shipping number of the product they bought, then go to the company's website from where they bought the product. Then they have to go to the delivery service's website to enter the shipping number.

No wonder many customers prefer asking a customer support agent to provide their product's shipping status.

A better solution overall would be to employ a chatbot. Customers can simply enter their product's shipping ID there and get a status update. popular European fashion brand, uses this feature in its chatbot use cases to provide instant order tracking for its customers – right after they have made a purchase. This frees their customer support team to cater to those customers who need support for more complex problems.

7. Chatbots for Effortless Feedback Collection

Traditional customer surveys suffer from low engagement due to cumbersome processes - requiring users to click links, fill lengthy forms, and type responses. Studies show email surveys average just **10% open rates** and **<5% completion rates**.

Chatbots transform feedback collection by making it:

Frictionless

- Customers participate via simple taps in messaging apps
- No forms to fill - just select pre-set options
- Instant access without redirects

Engaging

- Conversational format feels natural
- Visual elements (GIFs/images) boost interaction
- Takes <1 minute to complete

Actionable

- Real-time feedback analysis
- Instant follow-up for negative scores
- Seamless integration with CRM systems

Proven Results:

- **3-5x higher completion rates** vs email surveys
- **34% reduction** in customer frustration
- **Faster insights** to improve products/services

Example: A retail chatbot asks:

"How was your experience today?"

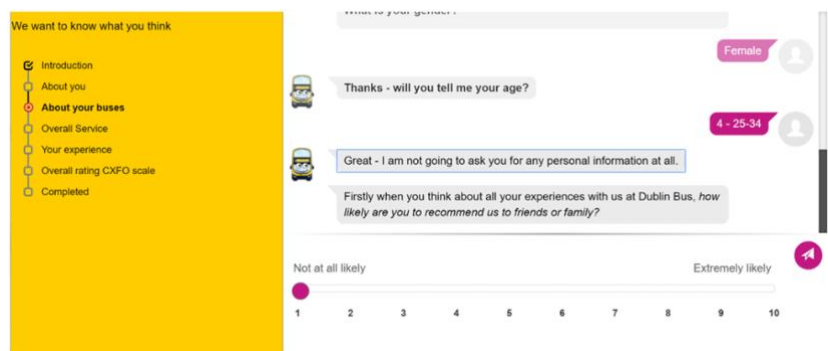
[😊 Great] [😐 Okay] [😞 Poor]

→ If "Poor", it immediately routes to human support.

Why It Works:

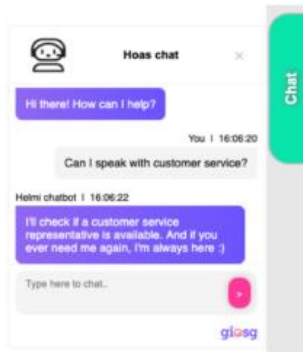
- Respects customers' time
- Feels like a conversation, not a chore
- Delivers instant value with quick resolution

Best For: Retail, SaaS, hospitality - any business needing real-time customer insights without survey fatigue.



Feature	Chatbot	Email Survey
Completion Rate	40-60%	<5%
Time to Complete	<60 sec	3-5 min
Customer Effort	Tap options	Type/click
Response Time	Instant	Days/weeks

8. Chatbots assign customer requests to support teams & Smart Ticket Routing with Chatbots



Problem:

Complex customer issues often get stuck in:

- Long email chains (24+ hour wait times)
- Endless phone transfers ("Let me connect you to...")

Chatbot Solution:

Instant Agent Connection

- Customer types "I need human help" Bot checks agent availability in real-time

- Seamlessly transfers chat + pre-fills ticket details

Intelligent Routing

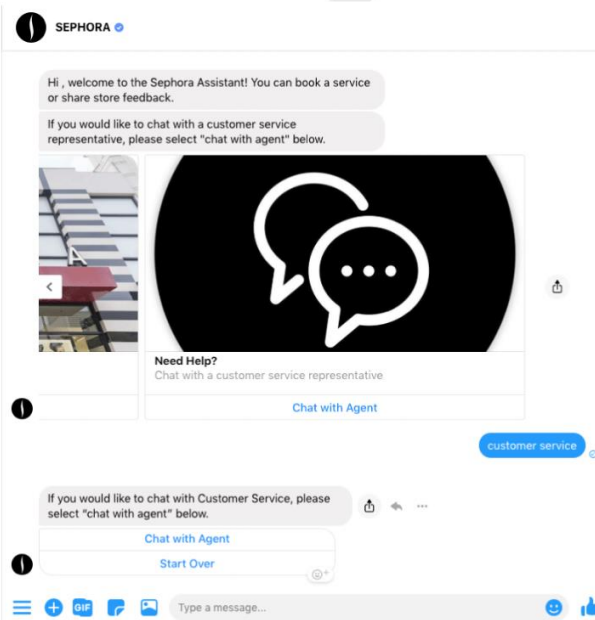
- Assigns queries by department (billing/tech support)
- Prioritizes urgent cases (VIP customers/payment issues)
- Provides agents with full chat history

Real Results:

- HOAS (Helsinki Housing): Handles **59% queries** via bot, routes rest to humans
- Sephora: One-click email capture for specialist requests
- **40% faster** resolution vs traditional ticketing

How It Works:

1. Customer asks bot for help
2. Bot attempts resolution first
3. If stuck → "Let me connect you to Sarah (Billing Dept)"
4. Agent receives:



- Customer name
- Issue summary
- Previous chat logs

Key Benefits:

- ✓ No channel-switching (chat stays in one thread)
- ✓ Agents get full context before replying
- ✓ 24/7 first-response, even when team's offline

Best For:

- E-commerce (high-ticket items)
- SaaS (technical support)
- Hospitality (urgent requests)

Pro Tip: Add a satisfaction survey after human interactions to close the feedback loop.

9. Chatbots Help Simplify Pricing – Key Notes

Problem:

Pricing pages can overwhelm customers with multiple plans and feature comparisons.

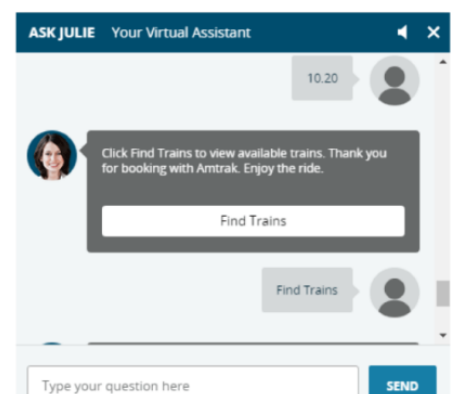
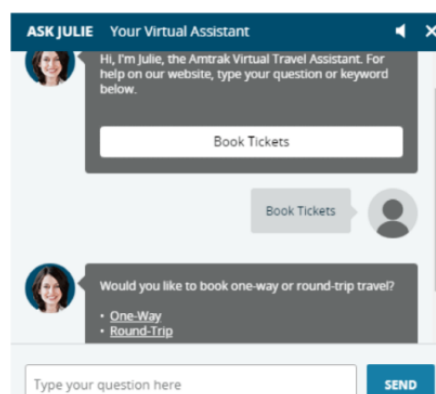
Impact:

Confused prospects may leave without buying and might hesitate to ask for help.

Solution:

Place a **chatbot** on the pricing page to:

- Gently offer assistance
- Help users compare plans
- Guide them to the right product/service tier



- **Real-World Example – Amtrak (USA & Canada):**

- Deployed "Julie" chatbot to help users book train tickets and choose routes.
- Results:
 - 25% increase in bookings
 - 50% increase in user engagement
 - Handles ~5 million questions/year

- **Takeaway:**

Any company with complex pricing can use chatbots to **simplify decisions**, increase conversions, and improve customer experience.

10. Chatbots for Engaging Product Promotion

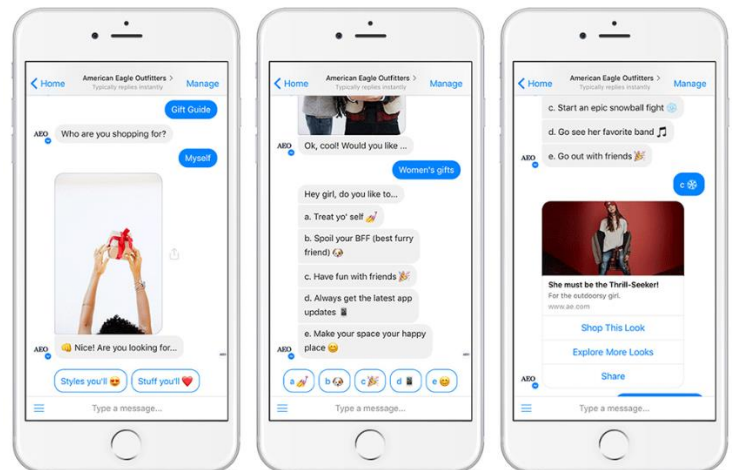
Why It Works:

Traditional ads and forms feel impersonal. Chatbots make marketing **fun and interactive** by:

- **Using humour & pop culture** (memes, GIFs, emojis)
- **Offering quizzes & games** ("Which product suits your style?")
- **Rewarding participation** (discounts, free samples)

Real Example: American Eagle Outfitters

- Targets **teen/young adult shoppers** with playful chats
- Asks style questions: *"Beach vibes or city chic?"*
- Recommends outfits + **offers 15% off quiz completion**
- Boosts engagement + conversions



How to Implement It:

1. **Quiz-Based Promotions**
"Take our 30-second quiz – get a surprise discount!"
 → Recommends products + unlocks coupon
2. **Interactive Storytelling**
"Help Emma pick an outfit for her date! 🍷 or 🍷?"
 → Shows matching products
3. **Viral Content Sharing**
"Tag 3 friends who need this meme-worthy deal! 🍷"

Key Benefits:

Higher engagement vs static ads

Personalized recommendations through conversations

More leads with incentive-driven quizzes

Best For:

- Fashion brands
- Beauty products
- Food & beverage
- Entertainment (movies, games)

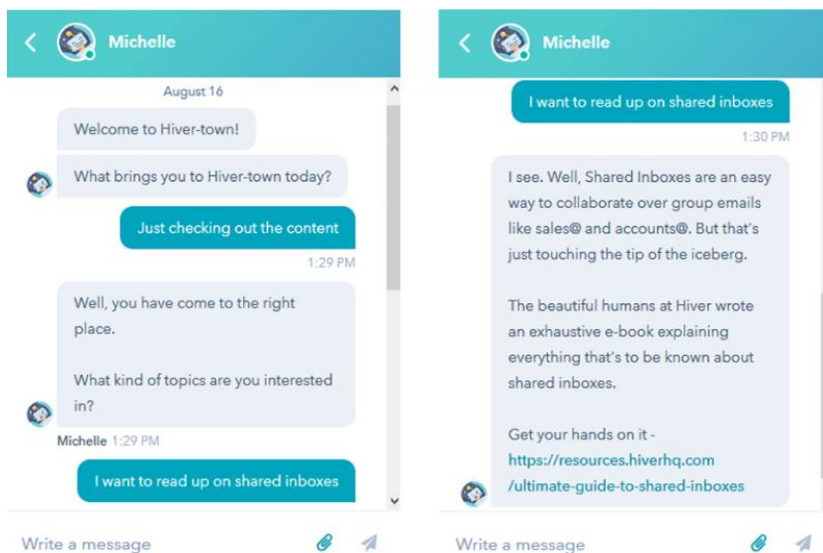
Pro Tip: Use **user-generated content** (UGC) in bot chats – *"Here's how others styled this jacket!"*

Want a ready-to-use quiz bot script? [Ask for examples]

Why Customers Love It:

- Feels like texting a friend, not being "sold to"
- Gets instant rewards (no forms, no wait)
- Makes shopping **an experience, not a task**

11. Chatbots for Smarter Lead Generation

**The Problem:**

Asking for a sale too soon scares away visitors. Traditional pop-ups and forms feel intrusive, leading to high bounce rates.

The Chatbot Solution:

A friendly, conversational approach that:

1. **Welcomes visitors** naturally ("Hi there! Need help finding something?")
2. **Offers value first** – free guides, demos, or tips
3. **Qualifies leads** by understanding their needs
4. **Collects emails organically** during helpful conversations

How It Works (Example: Vainu's Chatbot)**Engagement Hook:**

"Want to boost your sales with data insights?"

1. **Qualifying Questions:**

- "What's your biggest marketing challenge?"
[🌱 Lead generation] [📈 Conversion] [🔍 Data]

2. **Value Exchange:**

"Get our free 'B2B Growth Playbook' – just share your email!"

3. **Seamless Follow-Up:**

- Adds leads to CRM
- Triggers personalized email sequences

Key Benefits

Higher conversion rates (vs. static forms)

24/7 lead generation without human agents

Warmer leads – visitors self-qualify through chats

Proven Results:

- **Vainu** increased qualified leads by **40%** with their chatbot
- **63% of people** prefer chatbots over forms for sharing contact info (*Drift*)

Best Practices

1. **Start with help, not sales:**
"Need tips on [industry problem]? I've got a free guide!"
2. **Use button-based menus** for faster replies.
3. **Integrate with CRM** (HubSpot, Mailchimp) to automate follow-ups.

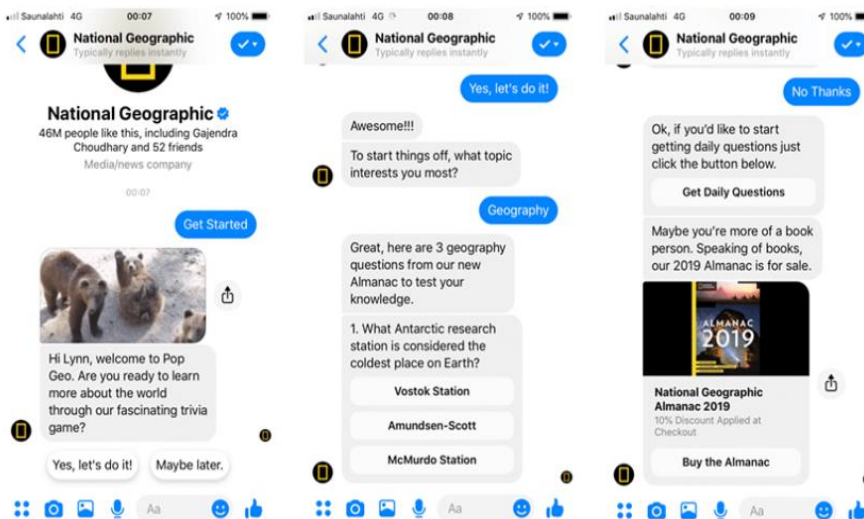
Want a script template for your industry? [\[Ask here\]](#)

Why It Works:

- Builds trust before asking for information
- Feels like a conversation, not a pitch
- Fits perfectly in the **top/middle of the marketing funnel**

12. Chatbots for Quizzes, Promotions & Contests

Why It Works:



People love **interactive content** and **rewards**.

Chatbots make promotions:

Highly visible – Pops up on your website/app

Engaging – Uses quizzes, trivia, or spin-the-wheel games

Lead-generating – Collects emails effortlessly

How Businesses Use It

1. Discount Quizzes

"Which [Product] Matches Your Style? Take the Quiz – Get 15% Off!"

→ Recommends products + unlocks coupon

2. Giveaway Contests

"Tag 2 friends & enter to win a free [Product]! 🎁"

→ Shares on social media for viral growth

3. Trivia Games (Like NatGeo)

"Test your knowledge! Score 5/5 & get our eBook at 10% off."

Key Benefits

Boosts engagement (5x higher than static ads)

Collects emails naturally during fun interactions

Drives sales with post-quiz discounts

Example: National Geographic

- Used a **trivia chatbot** to promote their Almanac eBook
- Offered **10% discount** to participants
- Increased **conversions + email signups**

How to Implement

1. **Choose a Prize** (Discount, free sample, eBook)
2. **Design a Simple Quiz** (3-5 questions max)
3. **Set Up Auto-Responses**
 - "Thanks for playing! Here's your 15% code: CHATWIN"
4. **Follow Up**
 - Send the prize + nurture leads via email

Best For:

- E-commerce stores
- Publishers (ebooks/courses)
- Restaurants (promo codes)

Want a ready-to-use quiz template? [Ask here]

(Word count: 150)

Why Customers Love It:

- Feels like a game, not an ad
- Instant rewards = instant satisfaction
- Easy to share (virality built-in)

13. Chatbots for Personalized Product Discovery

Why It Works:

Customers **buy more** when recommendations feel **tailored** to them. Chatbots excel at this by:

Guiding shoppers through curated selections

Highlighting new arrivals based on preferences

Reducing decision fatigue with smart filters

How Brands Use It (Example: MVMT Watches)

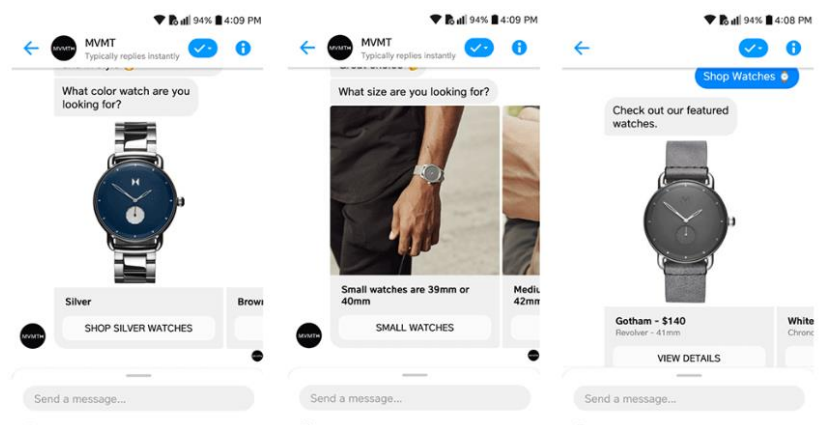
1. **Style Quiz**

"What's your vibe? 🌈 Minimalist | 🌈 Bold | ❤️ Classic"
2. **Personalized Picks**

→ Shows 3 best-matched watches with images
3. **New Arrivals Tease**

"Just in! The 'Chrono Eclipse' – limited stock!"
- 4.

Result:



- Higher conversions (vs. generic product grids)
- Increased average order value (AOV)

Key Features

Visual Recommendations (Product carousels + high-res images)

Smart Questions (Budget/colour/usage preferences)

Upsell Triggers ("Pair with matching sunglasses?")

Best For:

- Fashion brands
- Electronics retailers
- Subscription services

Pro Tip: Add **social proof** in chats:

"🔥 82% of shoppers like you picked this watch!"

14. Chatbots for Strategic Upselling & Cross-Selling

Why It Works:

70% of customers welcome post-purchase engagement (*HBR*), but timing is critical. Chatbots deliver **personalized offers when engagement is highest** – during active product exploration or checkout.

Key Tactics:

1. **Upsell Premium Versions**
"Upgrade to deluxe for 30% more features – just ₹200 extra!"
2. **Downsell Budget Options**
"Try our Essentials Plan at 40% savings."
3. **Cross-Sell Complementary Items**
"Pair this with our best-selling accessory (15% off combo)!"

HelloFresh's Success:

- Uses **chatbot-exclusive codes** (FRESHBOT25)
- Suggests meal add-ons during ordering
- Tracks conversions directly to chatbot ROI

Implementation Guide:

- **Trigger offers mid-conversation** when interest peaks
- **Use cart data** for hyper-relevant suggestions
- **Add urgency:** "Deal expires in 1 hour!"

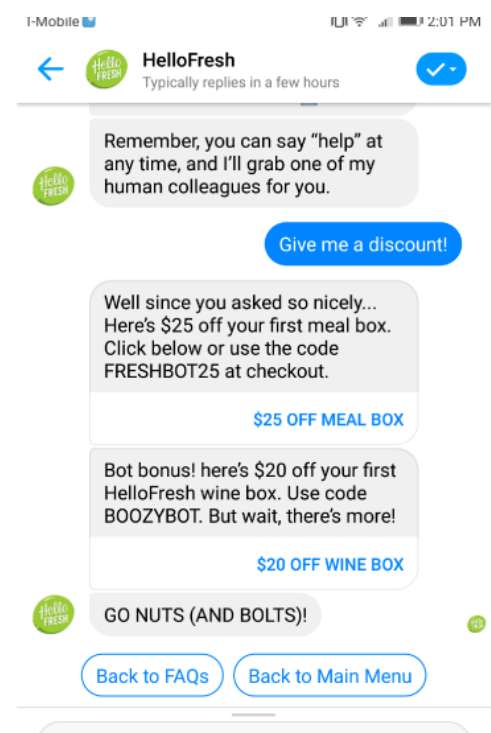
Results You Can Expect:

- 📈 20-30% higher average order value
- 🛒 15% more completed purchases
- Clear attribution of chatbot-driven sales

Pro Tip: Combine with **social proof**:

"92% of buyers added this to their order!"

Best For: E-commerce, SaaS, subscription services.



15. Chatbots for Guided Product Discovery

The Problem:

Too many choices overwhelm customers, leading to **decision paralysis** and abandoned carts.

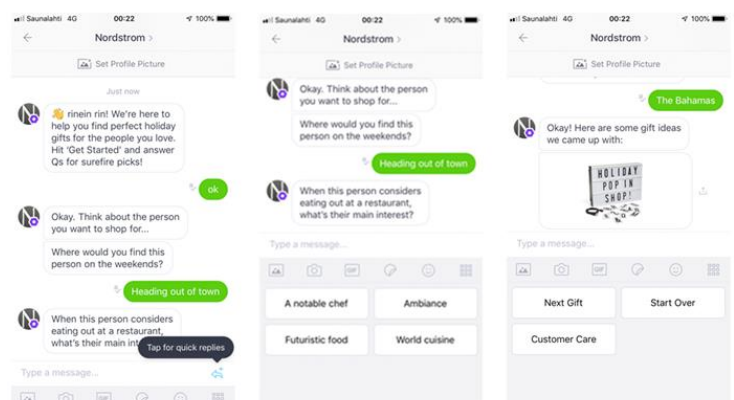
The Chatbot Solution:

Step-by-step product filtering that:

1. **Simplifies choices** with visual categories (images/videos)
2. **Asks qualifying questions** ("What's your budget?")
3. **Refines results iteratively** until perfect match

How It Works (Example: Nordstrom's Kik Bot)

1. **Category Selection:**
"What are you shopping for?"
[👗 Dresses] [👟 Sneakers] [👜 Bags]



2. **Style Preferences:**
"Choose a vibe:"
[🌟 Glam] [🧢 Casual] [❤️ Edgy]
3. **Personalized Showcase:**
→ Displays 3-5 tailored options with high-res images

Result:

- 50% fewer abandoned carts (*Nordstrom data*)
- Higher satisfaction (guided vs. overwhelming grids)

Key Features

Visual Menus (Images/GIFs/videos for each category)

Conversational Refinement ("Need this for a wedding or work?")

Instant Demo Links (Embedded product videos)

Best For:

- Fashion retailers
- Electronics brands
- Gift marketplaces

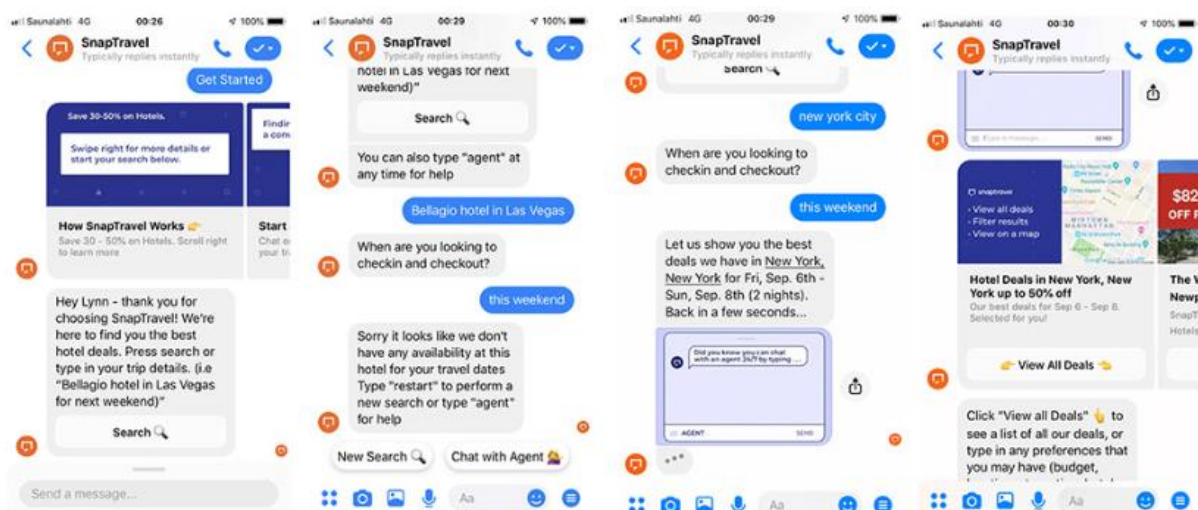
Pro Tip: Add "I'm Feeling Lucky" for indecisive shoppers → suggests a bestselling item.

Why Customers Love It:

- Feels like a personal shopper
- No endless scrolling
- Finds perfect fit faster

Want a template for your product catalogue? [Ask here]

16. Chatbots as 24/7 Personal Shopping Assistants



The

Problem:

Modern shoppers demand **instant, detailed product info** – but hiring human assistants 24/7 is costly and hard to scale.

The AI Solution:

Chatbots act as always-available personal shoppers that:

- **Share specs** (sizes/colours in stock) in seconds
- **Provide demos/guides** (videos, styling tips)
- **Recommend alternatives** if items are unavailable

How It Works (Example: H&M's Chatbot)

1. **Instant Product Lookup:**
"Is this dress in size M?" → Checks inventory in real-time.
2. **Style Advice:**
"How to style this jacket?" → Shows outfit ideas.
3. **Size Guidance:**
"I'm 5'9" – will these jeans fit?" → Recommends cuts.

Results:

- **90% faster** responses vs. email support
- **35% fewer** abandoned carts (*H&M data*)

Key Features**AI-Powered Search**

- Understands natural queries ("Show blue dresses under ₹3K")
- Visual Catalogue**

- Embeds carousels with images/videos
Seamless Handoff
- Escalates to humans for complex requests

Best For:

- Fashion/apparel brands
- Electronics retailers
- Beauty/personal care

Pro Tip: Add "**Notify Me When Back in Stock**" to capture leads.

Why Shoppers Prefer It:

- No wait times → instant gratification
- Gets **personalized** recommendations
- Accessible via WhatsApp/Instagram (no app needed)

Want a pre-built assistant script? [Ask here]

17. Chatbots as 24/7 Sales Agents

The Challenge:

Most businesses can't afford round-the-clock sales teams, yet customers expect **instant service** – even after hours.

The Solution:

AI chatbots act as always-on sales reps that:

Showcase products with rich media (images/videos)

Close simple sales autonomously

Capture leads for follow-ups

Escalate smoothly to humans when needed

How It Works (Example: Snaptravel)

1. **Instant Booking:**
"Book a Bangkok hotel under ₹5K/night" → Shows curated options.
2. **Lead Nurturing:**
If no purchase → "Get alerts for flash deals! Share your email?"
3. **Seamless Handoff:**
Complex queries → Routes to live agents with full chat history.

Results:

-  **30% more bookings** after-hours
-  **50% lead capture rate** from abandoned chats

Key Features

Product Demos: Embedded videos/GIFs in chats

Conversational Forms: "What's your budget?" → Suggests matches

Appointment Booking: "Schedule a demo with our team!"

Best For:

- E-commerce stores
- Travel agencies (like Snaptravel)
- SaaS companies

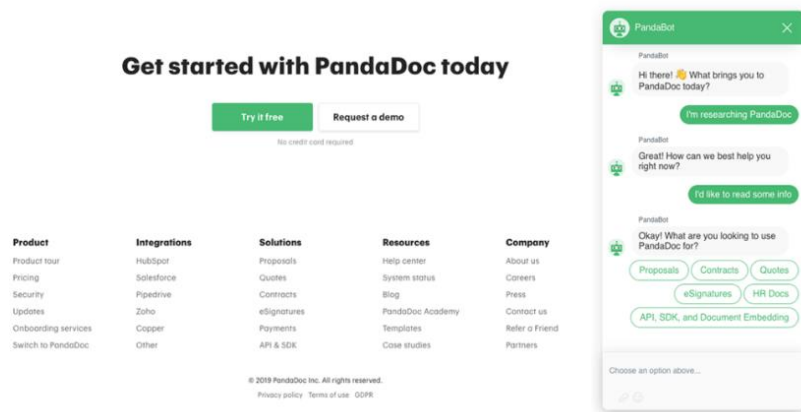
Pro Tip: Use **urgency triggers**:

"Only 2 units left at this price!"

Why Businesses Love It:

- Operates at **10% the cost** of human teams
- Converts **15-20% of after-hours traffic**
- Integrates with **WhatsApp/Messenger** (no app installs)

18. Chatbots for Instant Quotes & Estimates



The Problem:

Customers demand **quick pricing info** before buying, but manual quote generation bogs down sales teams.

The Chatbot Solution:

AI-powered quote engines that:

Pull real-time pricing from databases

Display product images + specs

Add "Buy Now" buttons for instant action

How It Works (Example: Staples)

1. **Customer Asks:**
"Price for wireless keyboards under ₹2K?"
2. **Chatbot Responds:**
 - Shows 3 options with images/prices
 - Adds: *"In stock! Buy now or compare more?"*
3. **Result:**
 - 80% faster quotes vs. email
 - 25% higher click-to-purchase rate

Key Features

Dynamic Pricing: Updates for discounts/inventory

Visual Quotes: Embeds product photos

CRM Integration: Logs leads for follow-ups

Best For:

- Office suppliers (like Staples)
- Insurance/loan providers
- Custom service businesses

Pro Tip: Add comparison tools:

"Here's how Plan A vs. Plan B stack up:"

Why Customers Prefer It:

- No waiting → instant transparency
- Visual proof builds trust
- One-click access to buy

Want a quote-bot template? [\[Ask here\]](#)

references

Application of computer vision in surveillance and security

<http://azure.microsoft.com>

chatbot use case in retail domain

<https://leaddesk.com/blog/chatbot-use-cases-25-real-life-examples/>

===== END OF TASK =====