

Create question answering solutions with Azure Language

1. Introduction

<https://learn.microsoft.com/en-us/training/modules/create-question-answer-solution-ai-language/1-introduction>

Introduction

Completed

- 2 minutes

A common pattern for "intelligent" applications is to enable users to ask questions using natural language, and receive appropriate answers. In effect, this kind of solution brings conversational intelligence to a traditional frequently asked questions (FAQ) publication. In this module, you will learn how to use Azure Language to create a knowledge base of question and answer pairs that can support an application or bot.

After completing this module, you'll be able to:

- Understand question answering and how it compares to language understanding.
- Create, test, publish and consume a knowledge base.
- Implement multi-turn conversation and active learning.
- Create a question answering bot to interact with using natural language.

2. Understand question answering

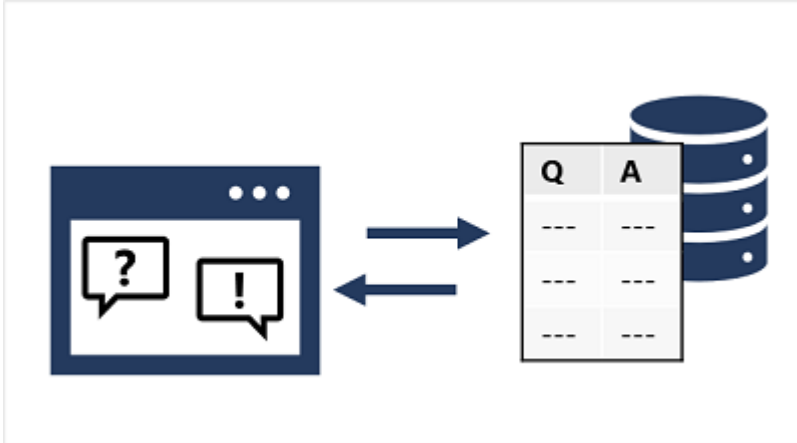
<https://learn.microsoft.com/en-us/training/modules/create-question-answer-solution-ai-language/2-understand-question-answer-capability>

Understand question answering

Completed

- 2 minutes

Azure Language includes a *question answering* capability, which enables you to define a *knowledge base* of question and answer pairs that can be queried using natural language input. The knowledge base can be published to a REST endpoint and consumed by client applications, commonly *bots*.



The knowledge base can be created from existing sources, including:

- Web sites containing frequently asked question (FAQ) documentation.
- Files containing structured text, such as brochures or user guides.
- Built-in *chit chat* question and answer pairs that encapsulate common conversational exchanges.

Note

The question answering capability of Azure Language is a newer version of the **QnA Service**, which still exists as a standalone service. To learn how to migrate a QnA Maker knowledge base to Azure Language, see the [migration guide](#).

3. Compare question answering to Azure Language understanding

Compare question answering to Azure Language understanding

Completed

- 2 minutes

A question answering knowledge base is a form of language model, which raises the question of when to use question answering, and when to use the *conversational language understanding* capabilities of Azure Language.

The two features are similar in that they both enable you to define a language model that can be queried using natural language expressions. However, there are some differences in the use cases that they are designed to address, as shown in the following table:

	Question answering	Language understanding
Usage pattern	User submits a question, expecting an answer	User submits an utterance, expecting an appropriate response or action
Query processing	Service uses natural language understanding to match the question to an answer in the knowledge base	Service uses natural language understanding to interpret the utterance, match it to an intent, and identify entities
Response	Response is a static answer to a known question	Response indicates the most likely intent and referenced entities
Client logic	Client application typically presents the answer to the user	Client application is responsible for performing appropriate action based on the detected intent

The two services are in fact complementary. You can build comprehensive natural language solutions that combine language understanding models and question answering knowledge bases.

4. Create a knowledge base

Create a knowledge base

Completed

- 2 minutes

To create a question answering solution, you can use the REST API or SDK to write code that defines, trains, and publishes the knowledge base. However, it's more common to use the [Language Studio](#) web interface to define and manage a knowledge base.

To create a knowledge base you:

1. Sign in to Azure portal.
2. Search for **Foundry Tools** using the search field at the top of the portal.
3. Select **Create** under the **Language Service** resource.
4. Create a resource in your Azure subscription:
 - Enable the *question answering* feature.
 - Create or select an **Azure AI Search** resource to host the knowledge base index.
5. In Language Studio, select your Azure Language resource and create a **Custom question answering** project.
6. Add one or more data sources to populate the knowledge base:
 - URLs for web pages containing FAQs.
 - Files containing structured text from which questions and answers can be derived.
 - Predefined *chit-chat* datasets that include common conversational questions and responses in a specified style.
7. Edit question and answer pairs in the portal.

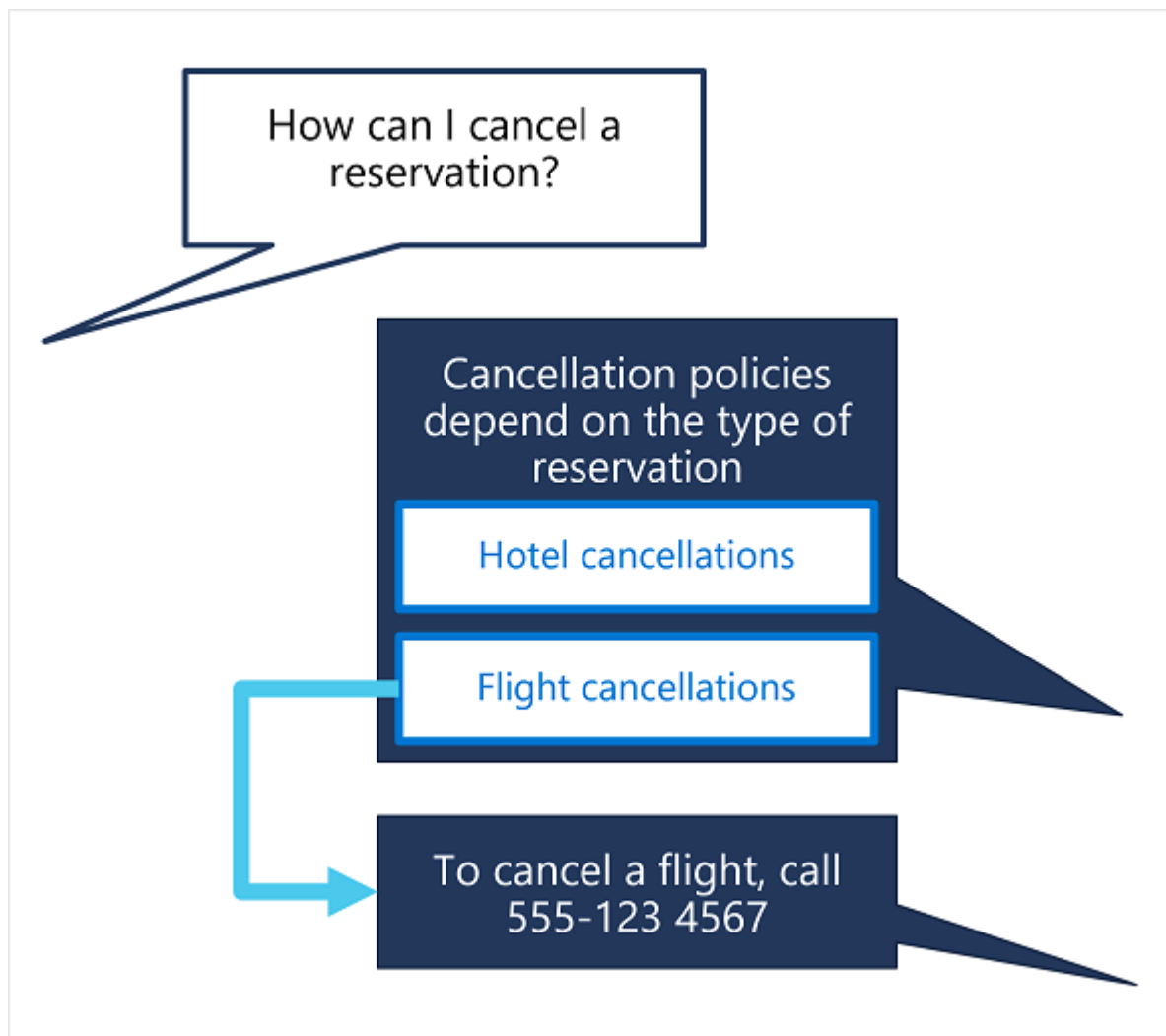
5. Implement multi-turn conversation

Implement multi-turn conversation

Completed

- 2 minutes

Although you can often create an effective knowledge base that consists of individual question and answer pairs, sometimes you might need to ask follow-up questions to elicit more information from a user before presenting a definitive answer. This kind of interaction is referred to as a *multi-turn* conversation.



You can enable multi-turn responses when importing questions and answers from an existing web page or document based on its structure, or you can explicitly define follow-up prompts and responses for existing question and answer pairs.

For example, suppose an initial question for a travel booking knowledge base is "How can I cancel a reservation?". A reservation might refer to a hotel or a flight, so a follow-up prompt is required to

clarify this detail. The answer might consist of text such as "Cancellation policies depend on the type of reservation" and include follow-up prompts with links to answers about canceling flights and canceling hotels.

When you define a follow-up prompt for multi-turn conversation, you can link to an existing answer in the knowledge base or define a new answer specifically for the follow-up. You can also restrict the linked answer so that it is only ever displayed in the context of the multi-turn conversation initiated by the original question.

6. Test and publish a knowledge base

<https://learn.microsoft.com/en-us/training/modules/create-question-answer-solution-ai-language/6-test-publish-knowledge-base>

Test and publish a knowledge base

Completed

- 1 minute

After you have defined a knowledge base, you can train its natural language model, and test it before publishing it for use in an application or bot.

Testing a knowledge base

You can test your knowledge base interactively in Language Studio, submitting questions and reviewing the answers that are returned. You can inspect the results to view their confidence scores as well as other potential answers.

Language Studio > Custom question answering > LearnFAQ - Edit knowledge base

Question answer pairs (150) Synonyms (0)

Search pairs

Learn more about certification.

You can learn more about certification on the [Microsoft certification page]...

I'm so bored of you	112
Swing and a miss.	1
Are you a woman?	68
That doesn't really apply to me.	1
Who's older?	47
I don't really have an age.	1
Will you ask me anything about myself	127
I'm a much better answerer than asker.	1
Do you have fingers?	74
I don't have the hardware for that.	1
Do you have a boss?	59
I'm here for you!	1
Can you jump rope?	209

Edit knowledge

0 unstructured sources and

Learn more about certification.

Source: Editorial

Answer

Edit answer

You can learn more about c (https://docs.microsoft.com

Alternate questions

Add an alternate question when questions should be as semantic

+ Add alternate question

Learn more about cert

Follow up prompts

Use follow up prompts to connect the client application to provide You can view all the connections

+ Add follow up prompt

Metadata (0)

Test

Response options

☐ Include short answer response

☐ Use deployed knowledge base

Show advanced options

Clear inspect

Hil

What is Microsoft Learn?

Inspect

We provide role and product-focused documentation, hands-on training, and certifications to help you get the most from Microsoft products.

Type your message and press enter

Deploying a knowledge base

When you're happy with the performance of your knowledge base, you can deploy it to a REST endpoint that client applications can use to submit questions and receive answers. You can deploy it directly from Language Studio.

7. Use a knowledge base

<https://learn.microsoft.com/en-us/training/modules/create-question-answer-solution-ai-language/7-consume-client-interfaces>

Use a knowledge base

Completed

- 3 minutes

To consume the published knowledge base, you can use the REST interface.

The minimal request body for the function contains a question, like this:

```
{
  "question": "What do I need to do to cancel a reservation?",
  "top": 2,
  "scoreThreshold": 20,
  "strictFilters": [
    {
      "name": "category",
      "value": "api"
    }
  ]
}
```

Property	Description
<code>question</code>	Question to send to the knowledge base.
<code>top</code>	Maximum number of answers to be returned.
<code>scoreThreshold</code>	Score threshold for answers returned.
<code>strictFilters</code>	Limit to only answers that contain the specified metadata.

The response includes the closest question match that was found in the knowledge base, along with the associated answer, the confidence score, and other metadata about the question and answer pair:

```
{
  "answers": [
    {
      "score": 27.74823341616769,
      "id": 20,
      "answer": "Call us on 555 123 4567 to cancel a reservation.",
      "questions": [
        "How can I cancel a reservation?"
      ],
      "metadata": [
        {
          "name": "category",
          "value": "api"
        }
      ]
    }
  ]
}
```



```
}  
]  
}
```

8. Improve question answering performance

<https://learn.microsoft.com/en-us/training/modules/create-question-answer-solution-ai-language/8-implement-active-learning>

Improve question answering performance

Completed

- 6 minutes

After creating and testing a knowledge base, you can improve its performance with *active learning* and by defining *synonyms*.

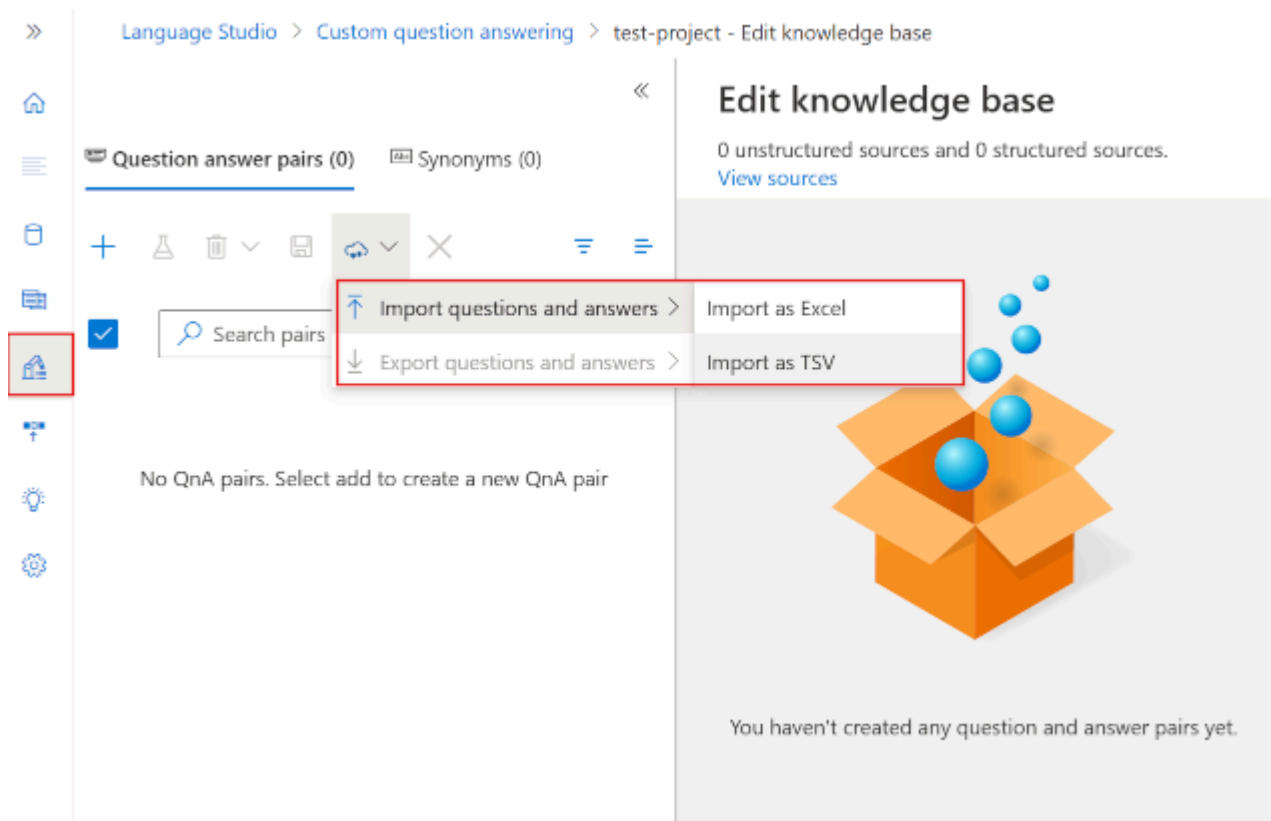
Use active learning

Active learning can help you make continuous improvements to get better at answering user questions correctly over time. People often ask questions that are phrased differently, but ultimately have the same meaning. Active learning can help in situations like this because it enables you to consider alternate questions to each question and answer pair. Active learning is enabled by default.

To use active learning, you can do the following:

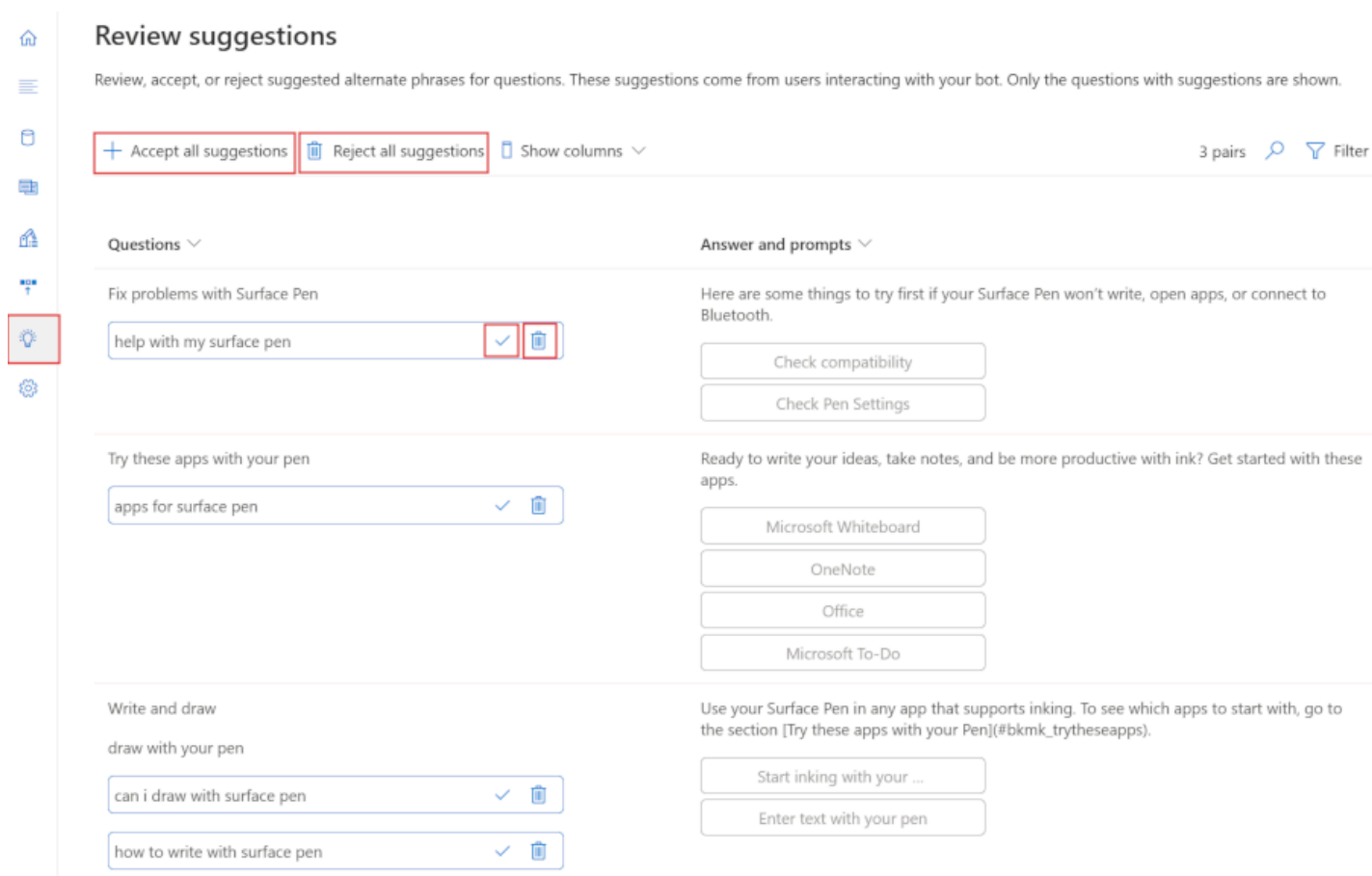
Create your question and answer pairs

You create pairs of questions and answers in Language Studio for your project. You can also import a file that contains question and answer pairs to upload in bulk.



Review suggestions

Active learning then begins to offer alternate questions for each question in your question and answer pairs. You access this from the Review suggestions pane:



You review, and then accept or reject these alternate phrases suggested for each question by selecting the checkmark or delete symbol next to the alternate phrase. You can bulk accept or reject

suggestions using the **Accept all suggestions** or **Reject all suggestions** option at the top.

You can also manually add alternate questions when you select **Add alternate question** for a pair in the Edit knowledge base pane:

The screenshot shows the 'Edit knowledge base' interface. On the left, a sidebar lists various knowledge base items. The main area displays the 'Replace Pen Tips' item, which includes a source, an answer, and a list of alternate questions. A red box highlights the '+ Add alternate question' button in the 'Alternate questions' section.

Note

To learn more about active learning, see [Enrich your project with active learning](#).

Define synonyms

Synonyms are useful when questions submitted by users might include multiple different words to mean the same thing. For example, a travel agency customer might refer to a "reservation" or a "booking". By defining these as synonyms, the question answering service can find an appropriate answer regardless of which term an individual customer uses.

To define synonyms, you use the REST API to submit synonyms in the following JSON format:

```
{
  "synonyms": [
    {
      "alterations": [
        "reservation",
        "booking"
      ]
    }
  ]
}
```

Note

To learn more about synonyms, see the [Improve quality of response with synonyms](#).

9. Exercise - Create a question answering solution

<https://learn.microsoft.com/en-us/training/modules/create-question-answer-solution-ai-language/10-exercise-create>

Exercise - Create a question answering solution

Completed

- 20 minutes

In this exercise, you use Azure Language to create your own question answering model.

Note

To complete this lab, you need an [Azure subscription](#).

Launch the exercise and follow the instructions.

[Launch Exercise](#)

Tip

After completing the exercise, if you've finished exploring Foundry Tools, delete the Azure resources that you created during the exercise.

10. Module assessment

Module assessment

Completed

- 3 minutes

11. Summary

<https://learn.microsoft.com/en-us/training/modules/create-question-answer-solution-ai-language/12-summary>

Summary

Completed

- 3 minutes

In this module, you have learned how to use the question answering capability of Azure Language to create a knowledge base of question and answer pairs that can support an application or bot.

Now that you've completed this module, you can:

- Understand question answering and how it compares to language understanding.
- Create, test, publish and consume a knowledge base.
- Implement multi-turn conversation and active learning.
- Create a question answering bot to interact with using natural language.

To learn more about the question answering capability of Azure Language, see the [Question answering documentation](#).