Topic 9
Implementing Bots

ST0249 (AIML) AI & MACHINE LEARNING

Learning Outcomes

- Implementing Bots
  - Training
  - Entities
  - Fulfilment

Training

Training

To make sure your agent matches user input as often as possible,

Provide your agent with enough data.

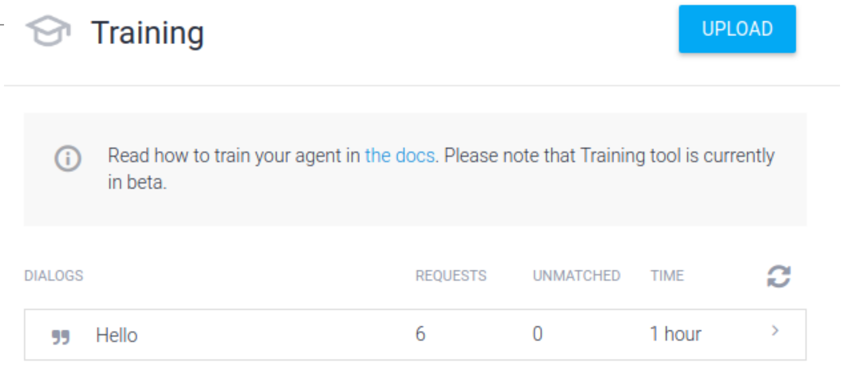
The greater the number of natural language examples in the Training Phrases section of Intents, the better the classification accuracy

When you save an intent, Dialogflow will begin training your agent with the new data you've added.

Until the training is complete, the updates may not be reflected in the agent.

How it works

As you and your users chat with your agent, you can access the conversation logs by clicking Training in the left menu.



Read how to train your agent in the docs. Please note that Training tool is currently in beta.

DIALOGS	REQUESTS	UNMATCHED	TIME
Hello	6	0	1 hour

How it works

Assigning a training phrase to an intent adds the example as a Training Phrases entry for that intent.

Whenever a user provides input, that input is recorded in the conversation logs along with the number of times that input was successfully or unsuccessfully matched to an intent.

These logs can help you discover gaps in your conversation setup.

The **Unmatched** number means the conversation contains the displayed number of interactions in which the user said something that wasn't matched to an intent

Upload Training Phrases

You can upload sample user inputs by clicking the UPLOAD button in the upper right hand corner.

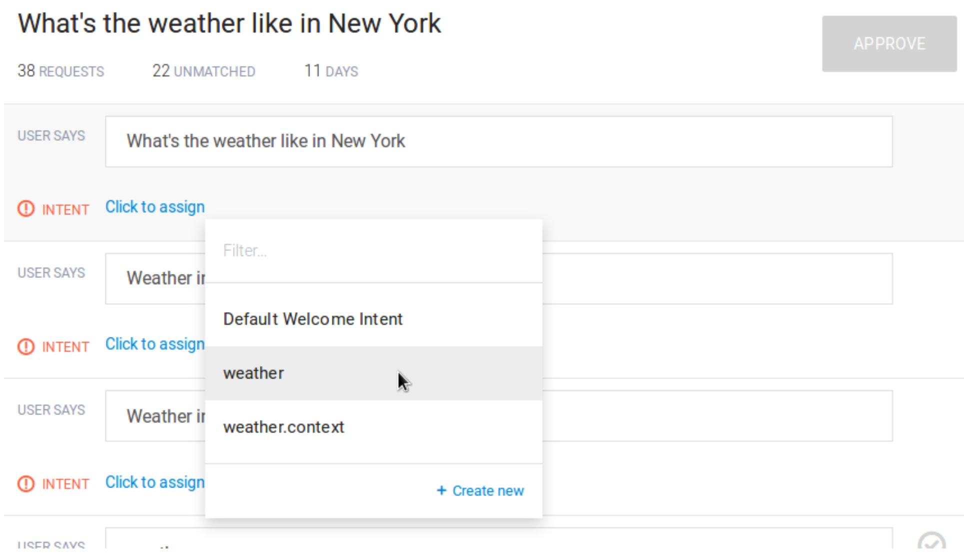
Training
UPLOAD

Handle unmatched inputs

Unmatched inputs are marked by an exclamation mark error_outline. You can assign unmatched inputs to intents in two ways:

Add inputs to one of the existing intents. When you click on Click to assign you'll see a list of existing intents to assign the unmatched phrase to. This will add the phrase to the intent's training phrases.

Create a new intent with this input.



What's the weather like in New York

38 REQUESTS 22 UNMATCHED 11 DAYS

USER SAYS What's the weather like in New York

INTENT Click to assign

USER SAYS Weather i

INTENT Click to assign

USER SAYS Weather i

INTENT Click to assign

Intents

An intent represents a mapping between what a user says and what action should be taken by your software.

Try it out

Intent interfaces have the following sections:

Training Phrases

Action

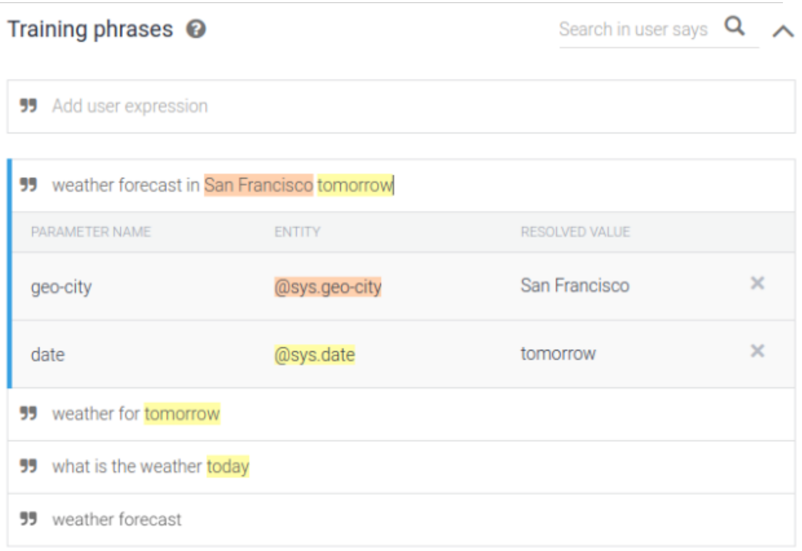
Response

Contexts

Annotations

Annotation is a process (and also the result of such process) of linking a word (or phrase) to an entity.

When you add examples to the 'User says' section, they are annotated automatically. The system detects the correspondence between words (or phrases) and existing developer and system entities and highlights such words and phrases. It also automatically assigns a parameter name to each detected entity.



Training phrases

Search in user says

Add user expression

weather forecast in San Francisco tomorrow

PARAMETER NAME	ENTITY	RESOLVED VALUE
geo-city	@sys.geo-city	San Francisco
date	@sys.date	tomorrow

weather for tomorrow

what is the weather today

weather forecast

Editing Annotations

You can edit the linked entity and the parameter name assigned to it in either the review window that opens when you click on the annotated example, or the parameter table of the ‘Action’ section.

You can do 3 type of changes:

- Assign a different entity to an annotated part of the example
- Edit the parameter name
- Delete the annotation (i.e., delete the link between the word and the entity).

Entities

Entities Introduction

Entities are powerful tools used for extracting parameter values from natural language inputs. Any important data you want to get from a user's request, will have a corresponding entity.

The entities used in a particular agent will depend on the parameter values that are expected to be returned as a result of the agent functioning. In other words, a developer does not need to create entities for every possible concept mentioned in the agent – only for those needed for actionable data.

Types of Entities

There are 3 types of entities: system (defined by Dialogflow), developer (defined by a developer), and user (built for each individual end-user in every request) entities. Each of these can be classified as mapping (having reference values), enum (having no reference values), or composite (containing other entities with aliases and returning object type values) entities.

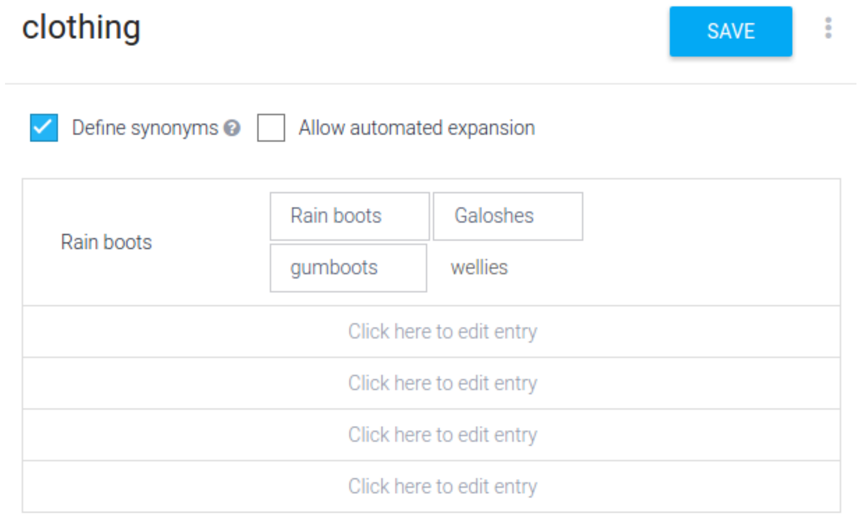
System Entities

System Entities are pre-built entities provided by Dialogflow in order to facilitate handling the most popular common concepts. Below are examples of the different types of system entities.

Try it

Developer Entities

You can create your own entities for your agents, either through web forms, uploading them in JSON or CSV formats, or via API calls.



clothing

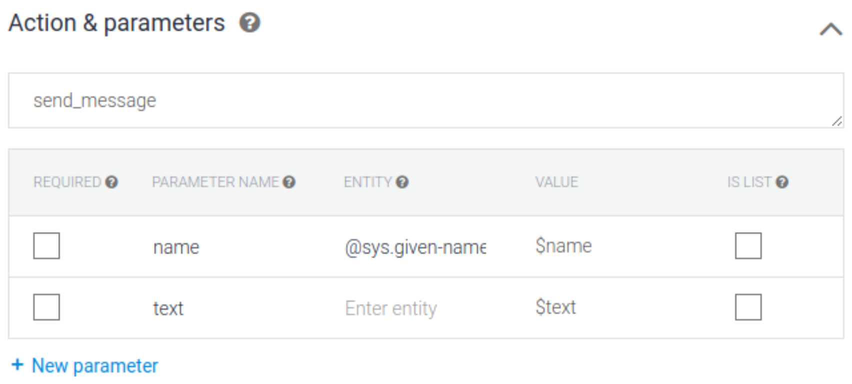
☑ Define synonyms ☐ Allow automated expansion

Rain boots	Rain boots, Galoshes, gumboots, wellies
Click here to edit entry	
Click here to edit entry	
Click here to edit entry	
Click here to edit entry	

Actions

An action corresponds to the step your application will take when a specific intent has been triggered by a user's input. Actions can have parameters for extracting information from user requests and will appear in the following format in a JSON response:

{"action":"action_name"}
{"parameter_name":"parameter_value"}



Action & parameters

send_message

REQUIRED	PARAMETER NAME	ENTITY	VALUE	IS LIST
☐	name	@sys.given-name	$name	☐
☐	text	Enter entity	$text	☐

+ New parameter

Parameters

Parameters are elements generally used to connect words in a user’s response, to entities. In JSON responses to a query, parameters are returned in the following format:

```
{“parameter_name”:”parameter_value”}
```

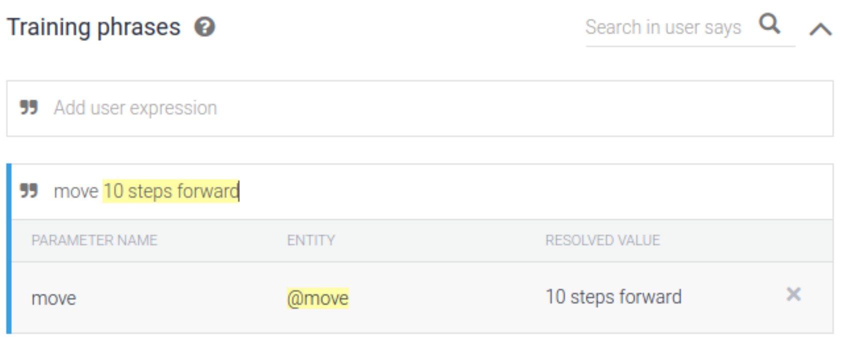
Parameters appear in two different areas. In the Training Phrases section, parameters related to known entities will be highlighted (annotated) after an example is added. Clicking on an annotated word in an example, will reveal a table with data on the chosen entity.

A table of the parameters in the intent, is located under the Action & Parameters section of the intent page. This table represents all of the parameters used throughout all of the Training Phrases examples in the current intent.

Defining Parameters

Automatic (Default)

When you enter an example in the Training Phrases section of an intent, words that correspond to system and developer entries will be highlighted (annotated), and parameters will automatically appear in the parameter table.


Training phrases

Search in user says

Add user expression

move 10 steps forward

PARAMETER NAME	ENTITY	RESOLVED VALUE
move	@move	10 steps forward

Manual

There are some cases when you may need to define or edit parameters and their values manually. For example, if your application controls a robot’s step motor, you may want to use fixed values for the parameters. You need to enter these values manually.

The Parameter Table

Required - Checking this option will mark the parameter as required, which means the parameter is needed to perform the action in the intent. When this option is enabled, a Prompts column will be revealed

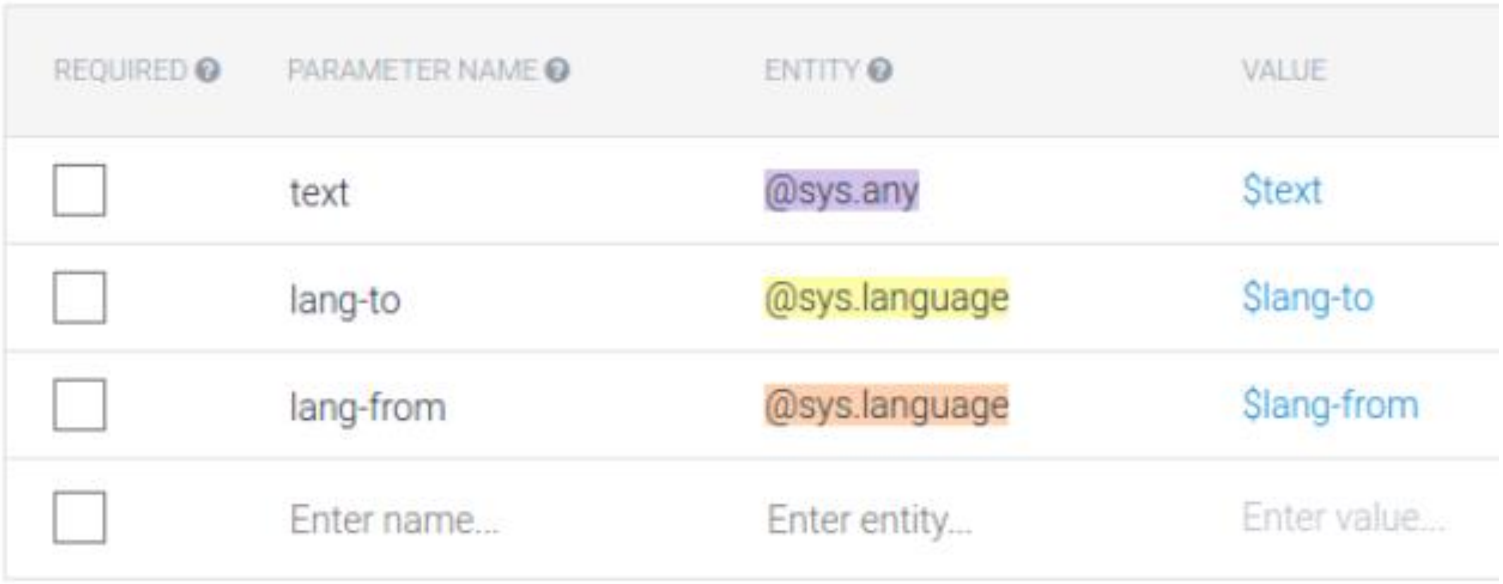
Parameter Name- This property defines how the parameter will show up in the JSON response: {“parameter_name”:”parameter_value”}

Entity. This property is the entity (system or developer) that is linked to the parameter. The linked entity can be changed by clicking on the current property and choosing a new entity from the searchable list.

Value. This property is the value assigned to the parameter. This is how the value will show up in the JSON response

{“parameter_name”:”parameter_value”}

Parameter Table

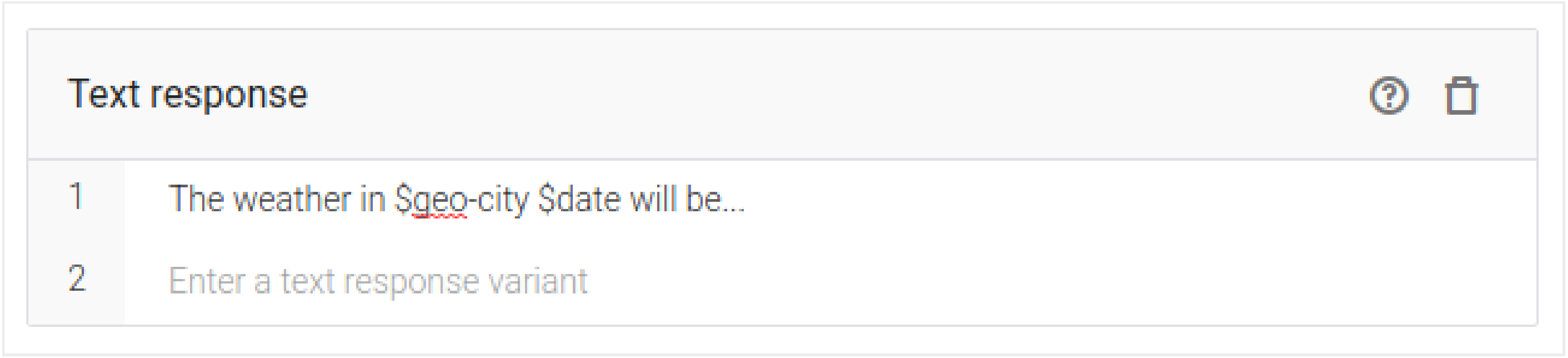


REQUIRED	PARAMETER NAME	ENTITY	VALUE
☐	text	@sys.any	$text
☐	lang-to	@sys.language	$lang-to
☐	lang-from	@sys.language	$lang-from
☐	Enter name...	Enter entity...	Enter value...

Referring to Parameter Values in Responses

To reference parameter values in the responses, use the following format:

$parameter_nam


Text response

1 The weather in $geo-city $date will be...

2 Enter a text response variant

Contexts

Contexts represent the current context of a user's request. This is helpful for differentiating phrases which may be vague or have different meanings depending on the user’s preferences, geographic location, the current page in an app, or the topic of conversation.

For example, if a user is listening to music and finds a band that catches their interest, they might say something like: “I want to hear more of them”. As a developer, you can include the name of the band in the context with the request, so that the agent can use it in other intents.

Try it

Adding Contexts

To define and add a context to an intent, click in the Add input context or Add output context field and type the desired name of the context and hit Enter to commit it.

Lifespan

By default, contexts in intents expire after either five requests or ten minutes from the time they were activated. Contexts in Follow-up intents have a default lifespan of two requests. Intents that renew the context will reset the counter and clock to give an additional five requests and ten minutes

Extracting Parameter Values from Contexts

To refer to a parameter value that has been extracted in an intent with a defined output context, use the following format in the VALUE column: #context_name.parameter_name

Fulfillment

Introduction

In order to add response logic or include the results of an API call in your agent's response, you need to setup fulfillment for the agent. This includes some basic JavaScript and setting up and hosting the files in a cloud service.

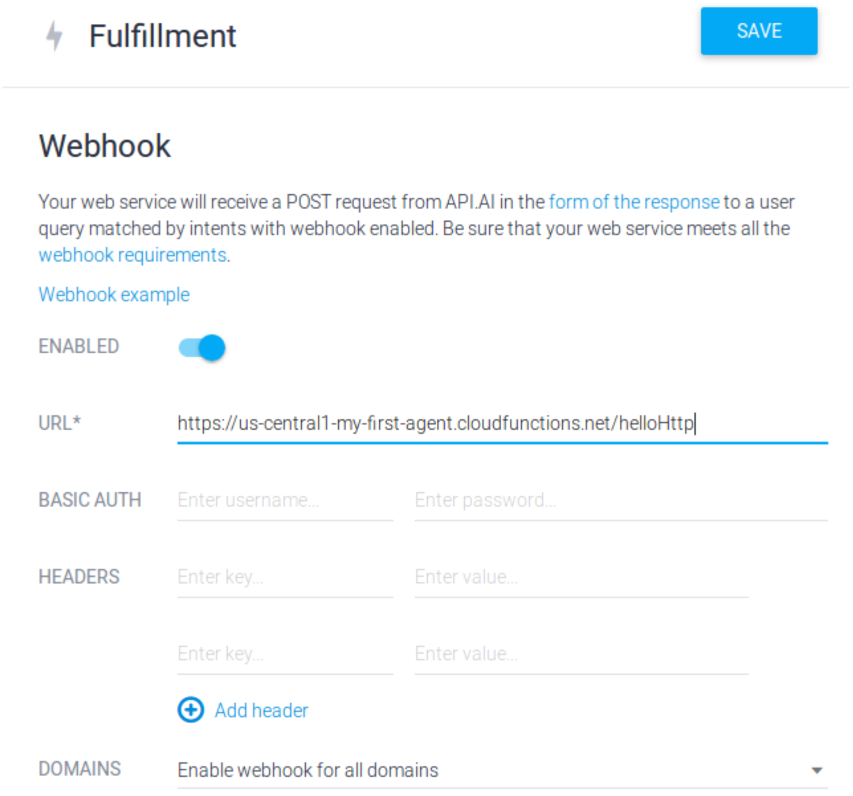
Webhook

In Dialogflow, make sure you're in the correct agent and click on Fulfillment in the left hand menu

Toggle the switch to enable the webhook for the agent

In the URL text field, enter the httpTrigger url you got when you deployed your function

Click Save



Webhook

Your web service will receive a POST request from API.AI in the form of the response to a user query matched by intents with webhook enabled. Be sure that your web service meets all the webhook requirements.

Webhook example

ENABLED

URL* https://us-central1-my-first-agent.cloudfunctions.net/helloHttp

BASIC AUTH Enter username... Enter password...

HEADERS Enter key... Enter value...

Enter key... Enter value...

Add header

DOMAINS Enable webhook for all domains

Summary

What we have learnt

- How to perform training for Bots (using DialogFlow)
- How to define your intents (using DialogFlow)
- How to define your own entities (using DialogFlow)
- How to customize Fulfilment actions (using DialogFlow)