Step by step: Dialogue with context.

In this section we will cover how to setup a more advanced intent step by step. We will create an intent that can be used with several different products. We will do the following steps.

- Create intents

- Create entities

- Create entity switch

- Create context switch

- Connecting your responses

- Testing the buikd

Using and understanding this method will let you build intents that can respond to different context and to keep this context throughout a customer journey.

Creating the intents.

1.Go to intents in the main menu

2.Click the plus sign, to create a new intent
3.Name your intent – "Need information"
4.Add the following training sentences
     1."I want to know more about"
     2."I want to have some information"
     3."Can I have the latest info regarding"
     4."I'm interested in knowing more about"
     5."I want to know more about"

    This will give you an intent with where the end users intention would be to get information about "something".

5.Train your new intent to your model and try it out.

    1.Click Train model
    2.Wait for the training to finish
    3.Go to Test Intents and try it out. Try to challenge the bot by not writing one of the existing sentences.

Creating entities

We now have working intent. The end user can now ask for information. But we also need to define what product or process we would like to be able to respond to.

In this example we will define a couple of vehicles as entities and try and see how the bot responds.

To do this, follow these steps.

1.Add the following products to the training sentences.

      1.I want to know more about aircrafts
      2.I want to have some information about trains
      3.Can I have the latest info regarding boats
      4.I'm interested in knowing more about cars
      5.I want to know more about bicycles

2.Define the products as entities by selecting the specific word with the mouse:

3.This will bring up a button

4.In the Manual Input Entity Variable field, enter "aircrafts" to define your entity as aircrafts.

5.Repeat this for all the products. This will give you the following results

6.Train your model and wait for the training to finish.

7. Test your result in "test intents"

Now we can see that the bot extracts the product that we specified in the training data.

I want some information about an aircraft

Best Intent Match: Need information

Confidence: 100%

Intents

Name	Confidence [%]
Need information	100

Entities

| Name	Value	Confidence [%]
aircrafts	aircraft	99

Create responses

Now we need to create our responses. The bot will need a response for every product that we have defined. Unless the information is the exact same for every product. Which is not the case in our example.

So now we will create the following workflows.

•NeedInformation – Cars
•NeedInformation – Aircrafts
•NeedInformation – Boats
•NeedInformation – Trains
•NeedInformation – Bicycles
•NeedInformation – Default

To make it easy for us. Create a workflow for each product using a "conversation message" like this.
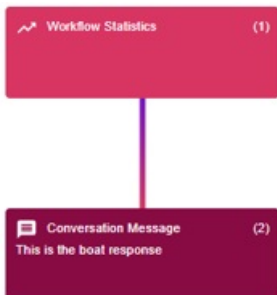
Input "This is the boat response", "This is the cars response" and so on..

The NeedInformation – Default, is special

The Default flow is created as a sort of fallback flow. If the bot picks up the intent, but does not pick up an entity. We will trigger the default flow.

Create the default flow like this.
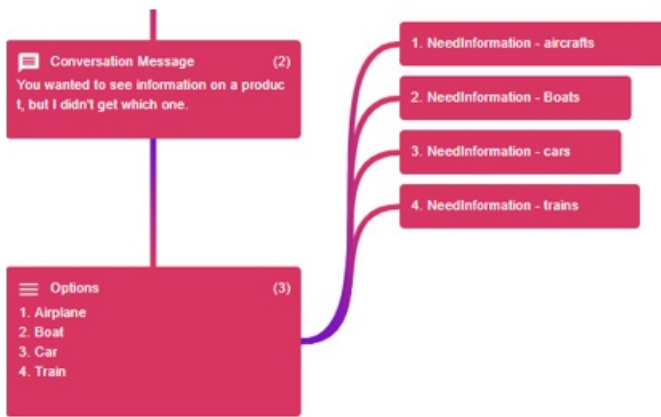
Conversation message saying: "You wanted information on a product. I didn't get which product"

Then buttons to give suggestions on products.

Create the entity switch

An intent can only hold one response. However, this respond can be a switch to branch out to specific responses depending on what entity the bot picked up.

So to make the bot respond depending on the product, we need to evaluate the product in an entity switch in the intent response.

As soon as an entity is extracted from the NLU. This entity will be stored in the bots memory. We can now use this to switch responses in the intent response.

1. Go to intents in the main menu.
2. Select the Need information intent
3. Go to the response tab

4. Add an "Entity Switch" as action.
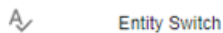
5. As Default, set your NeedInformation – Default
6. Add Checks to check if the bot has picked up an entity. Add one check for each defined entity.
7. For each check, add the corresponding workflow.

How to keep context in the dialogue

As you have read in the section "The concept of context", The variable "context" is the default variable where our latest context is stored. When the entity passes through an entity switch, that entity is first evaluated in the switch, then it's forgotten as an entity and stored as "context".

This means that "context" is what the bot is remembering from our last query.

So, to be able to be more generic in how we talk to the chatbot, we can now create a "context switch" as a workflow.
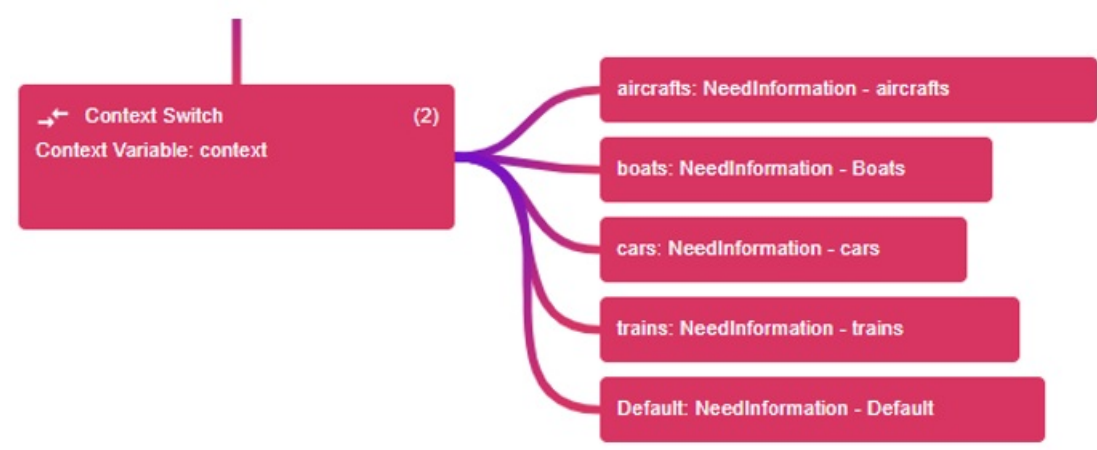
This Context Switch will make the bot look for a previous entity, now called "context".

If context is boats, we will trigger boats. If we do not have any context, we will trigger default. Just like the entiy switch.

When we have created the context switch. We will replace the default workflow ni out Entity switch. In this way we will first evaluate if we have an entity. If not, we will evaluate if we have a previous entity in context. If not, we will trigger the default flow.

By repeating these steps, we could now build a new intent where we ask about prices.

1. Go to intents in the main menu, Create a new intent called Prices
2. Add the following training sentences and entities.

Add Training Sentence

1. How much is it?

2. What does a [boat](boats) cost?

3. How much do I have to pay for an [aircraft](aircrafts)?

4. What's the price of a [train](trains)?

5. What to I pay for a [car](cars)?

3. Create workflows for the products.
    1. Prices – Cars
    2. Prices – Boats
    3. Prices – Aircrafts
    4. Prices – Trains
    5. Prices - Default
    6. Prices – ContextSwitch (Default flow should be Prices Default and manual input variable should be "context"

4. In the Prices intent response tab, create an entitySwitch with Prices – ContextSwitch as default and a Check for each entity.
5. Train the model, test it on your website or in Bankdemo site.
6. You should get this result.

"I want to have information on a boat."

Boats information is triggered. Entity is stored in "context" and forgotten as an entity.

Intent match (product information), Entity found "Boat"
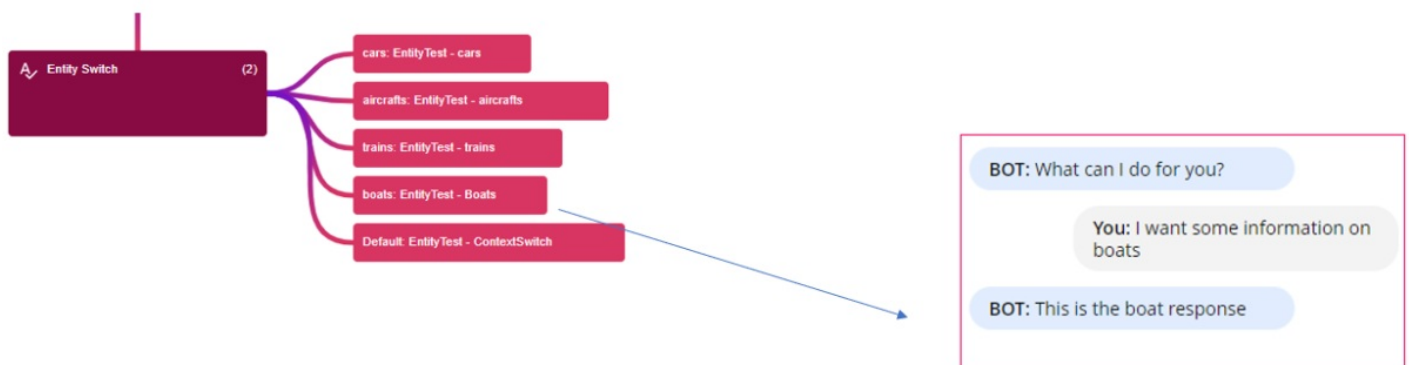
"How much does it cost?"

Default workflow in entity switch is triggered. Which is a context switch. context has been set to "boats" in previous query,

Intent match (product pricing), No entity found.