Optobot

Build bots with ease

February 2020

Optobot

Build bots with ease

- How to view a chatbot
- How to train **intents**
- How to train **conversation** flow
  - ➔ Variables
  - ➔ Responses
  - ➔ Options
  - ➔ Conditions

How to view a chat bot

- Conversations are always contextualized
- Each conversation have a start (an intent that is gathered from the user) and an end.
- Each conversation can be viewed as a path that the user can navigate through to reach a certain goal
- We like to view the conversation as a tree
- So each path start from a root and end in a leaf of the tree
- We can have multiple trees and multiple contexts.
    - A context is a group of certain conversations paths/trees that are related to each other.

How to train intents

- Intents (and entities) are mapping of user text to some predefined names
- We use Wit.ai for NLP handling and mapping of what the user says to intents
- Each intent has a name and a set of examples that is used to trail wit NLP engine
- Example:
  - [I:Greeting]
    - Hi
    - Hey
    - Hello
- Entities:
  - [E:wit$number]
    - 1000

OPTOMATICA
THE POWER TO PERFECT

How to train conversation flow

- Each conversation point is either a tree leaf or in the middle of the conversation (used to gather some sort of information or prepare for next point of interaction)
- Tree Leafs [**Q:**__name__], Middle nodes [**R:**__name__]
  - Each of these nodes have a unique name. Example: [Q:my_dialogue]
  - The name is used to navigate to when needed
  - Optionally, they can have an intent as [**R:**__name__**:**__intent__]
    - The intent specifies that optobot would go to this node when the user says something that match this intent. Example [R:my_dialogue:greeting]
    - My Dialogue would be stated when a greeting intent is captured

Variables

- Variables are definitions about some information that we want optobot to collect from the users. e.g. Age, name, intension, agreement, etc…

- Each variable will store user information when the user is asked about it and provide its answer.

OPTOMATICA
THE POWER TO PERFECT

Variables (cont'd)

- For example a variable can be **age** and its enquiry text can be “How old are you?”. Enquiry text is a response from optobot to ask about this variable
- Each variable has a type (number, location, string, etc..)
  - Variable types can be any type that wit supports
- Each variable has a storage_type. This can be either stored
  - **in_cache:** very temporarily. In current node only
  - **in_session:** until the user session expires
  - **timeseries:** an array stored every specific period of time
  - **timeseries_in_cache:** an array stored everytime regardless of how long since it was collected
  - **normal:** default (stored only once and never asked about again). Like age

Variables (cont'd)

- Another type of variable, that is not collected is fetched variables.
- Fetched variable just like variables have type and storage_type
- The start with [F:...] instead of [V:...]
- The they are followed by definition of how this variable is calculated.
  - It can be fetching something from a URL (get, post)
  - Or predefined built-in functions (like sum, subtract, division, multiply, power, weather, …)
- Both types have the function in the beginning and then the arguments (all separated by “,”)
- Examples:
  - [F:....] sum, a, b, c
  - [F:....] multiply, a, 2

OPTOMATICA
THE POWER TO PERFECT

Responses

- After defining the node [R], [Q] or [V], you can add the text/image that the bot would say when it reaches this node
- Example [Q:my_dialogue:greeting] **Hi, how can I help you ?**
- Content can be text or image as [Q:my_dialogue:greeting] **image@http://….**
- Multiple replies is also supported each in a separate line as follows

[Q:my_dialogue:greeting] **(response)** Hi, I'm optobot

(response) image@**http://www….**

(response) How can I help you ?

Options

- Options are quick replies that act like a suggestion to the user to choose among in a certain variable
- For example, when asking about something (v1) (do you want ... ?)
  - Options can be “**yes**” and “**no**”
- Options is stated as follows **[O]**yes, **[O]**no
- Whenever the user chooses one of the options the variable that these option belong to will store user data equal to this option
  - Earlier example. If the user chooses “yes” then v1 data for this user will be “yes”

Conditions

- Conditions are basically bot control flow.
- The decide how the bot respond in different scenarios.
- For instance, if we have a variable named age. If the age is less than 12, then we want to say something different than when the age is > 12
- To do this a condition [C:__destination_node__] age < 12
  and another condition [C:__another_destination_node__] age > 12
- destination_node is the name of the node to go to if this condition is true
- Operator can be >, >=, =, <, <=

Optobot

Any Question?