Smart Robotics project

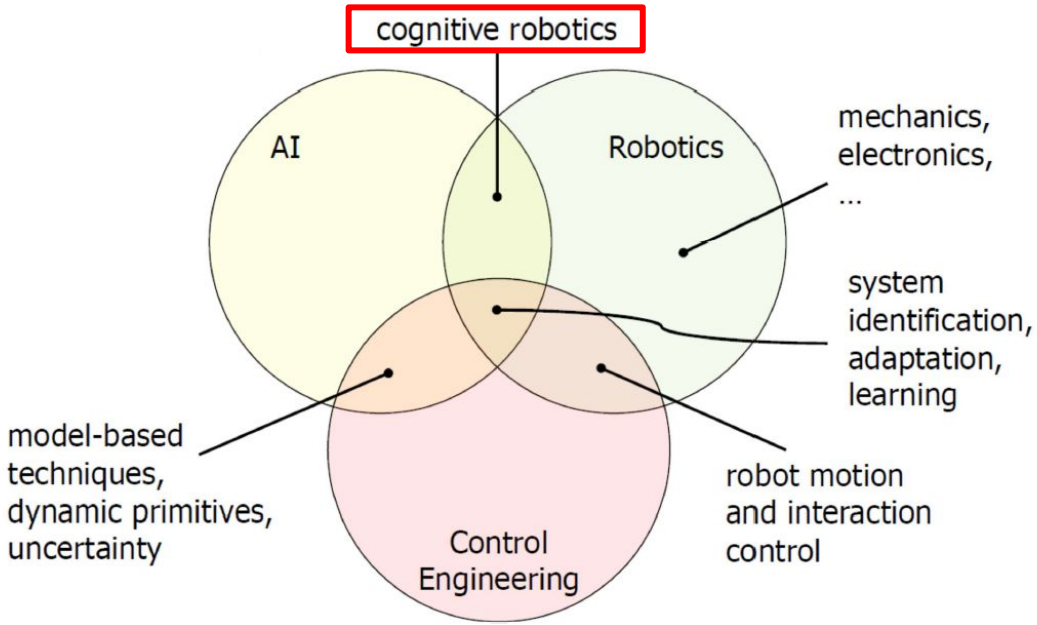
Baxter: the bartender understander

Matteo Mosconi

The starting point…

What is a robot?

Robots = embodied AI systems
Robotics = science of artifacts intelligently actuated and interacting with the real world



Baseline

Robots are nowadays considered embodied artificial intelligences. For sure, the most natural way in which us, as human beings, communicate with each other is our language. With this given, can we try to communicate with robots through language?

There are possible methods that try to map a phrase given as input, to a command that the robot will execute to perform a specific action. Some “traditional” techniques of this kind can involve the use of regular expressions (regex). These are unfortunately really limited, and are not suited for the irregularity of the language (i.e. the user would ends up saying or typing phrases always with the same “hardcoded structure” and length).

NLP (Natural Language Processing) is the field of artificial intelligence that deals with the language. In really recent years (May 2020, for what concern the first version of the paper talking about GPT-3) some really powerful models came out. They are capable of many tasks concerning language. In my case the focus point is related to the mapping between a phrase in plain language and a command (or a sequence of commands) that the robot will execute.

My work

The case of use in which I wanted to apply the key concepts already said is “bartender cocktail making”.

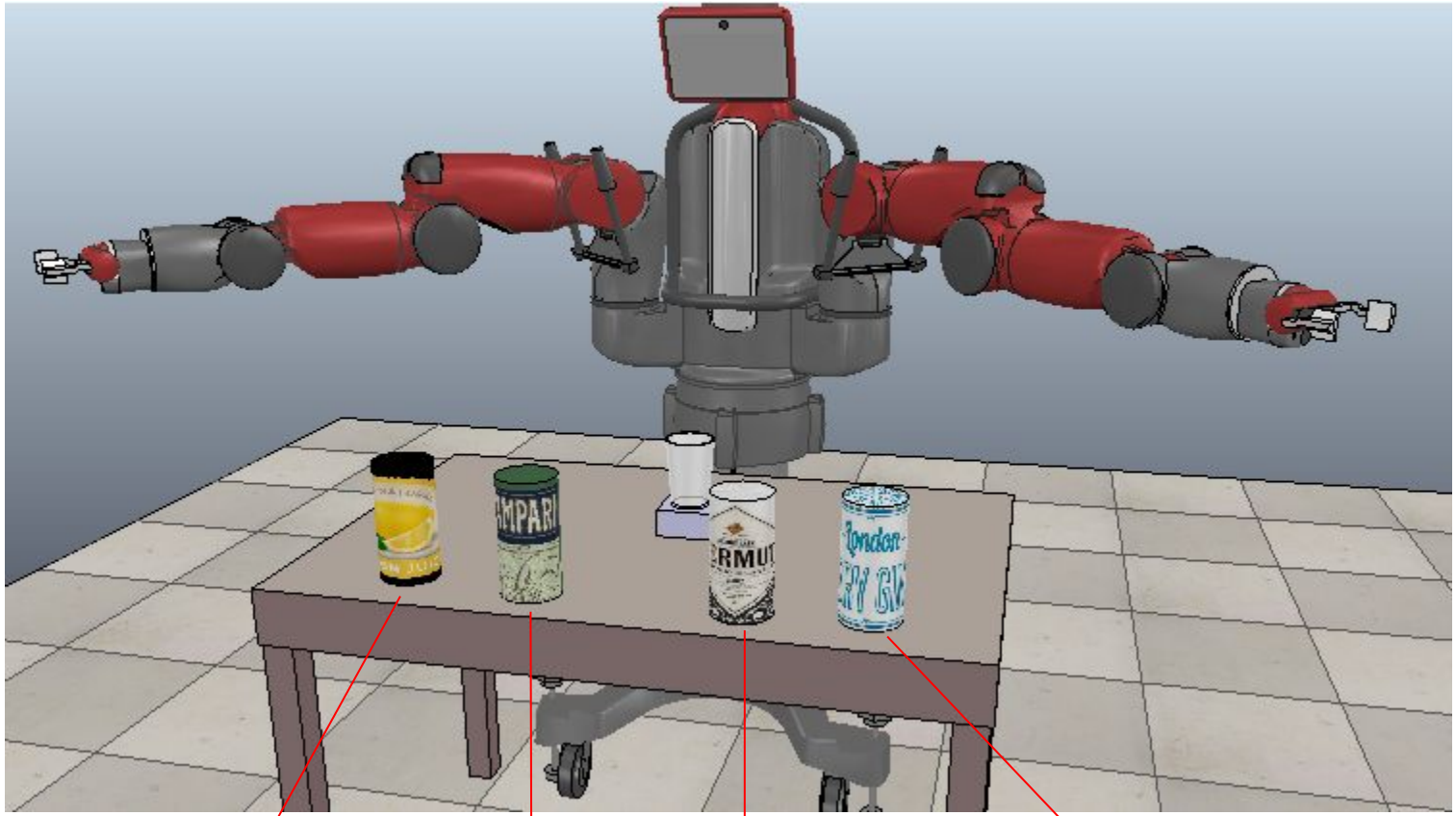
For this purpose I’ve used Baxter, an anthropomorphic robot with two arms that, between the robot available in the simulation environment that I’ve used, CoppeliaSim, is the most similar to a human bartender.

In order to achieve the understanding part I will use an implementation of GTP-3 by OpenAI, some details later.

The environment will be composed by a table, that is the working space, and four bottle of ingredients, that I simulate using four big cylinders. Finally a cup which will be the recipient in which Baxter will simulate the pouring action.

The input command in natural language can be virtually given with simple speech. But for the sake of stability (i.e. this task is highly dependent on the quality of the microphone and the noise) the textual mode is preferable.

The next slides will contain some visual examples along with some details.


lemon juice
Campari
vermut
gin

Pipeline and details

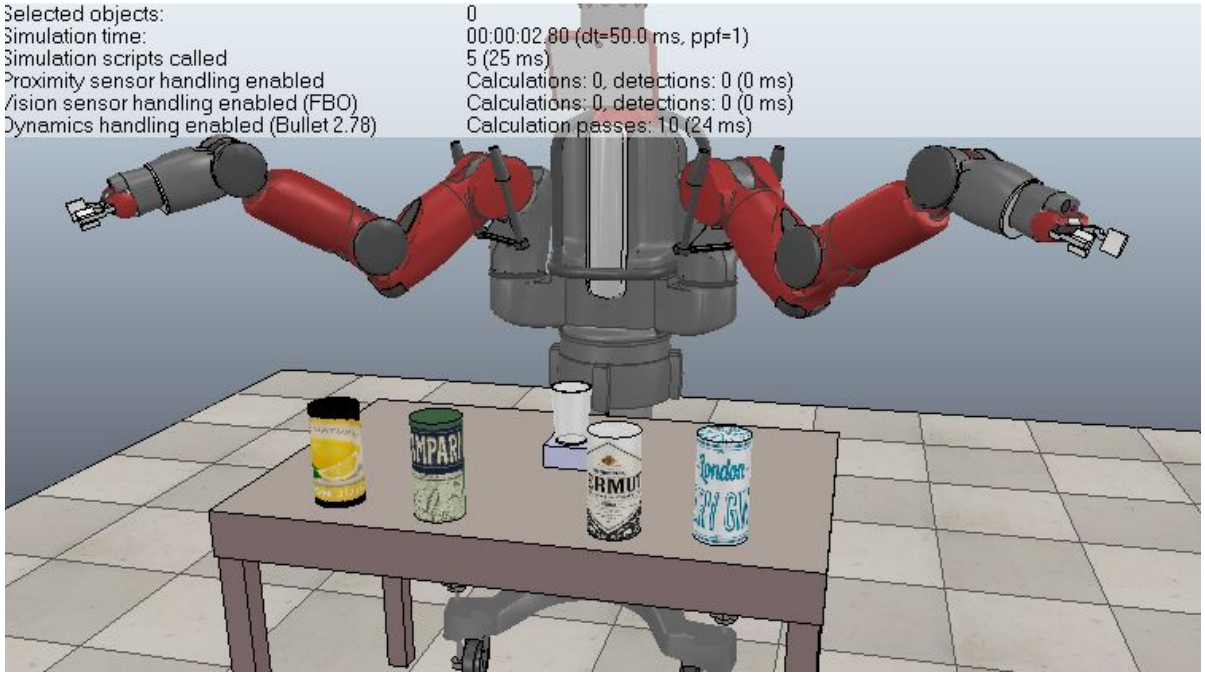
As the simple scheme above shows, the natural language input will be passed through GPT-3, that at its time will “produce” command output that will be executed in the CoppeliaSim simulation environment.

We can send orders for single or multiple ingredients cocktails, we can request a specific cocktail from a list of recipes available:

- Negroni: composed by gin, vermut and Campari
- Martini: composed by gin and vermut
- Gin lemon: composed by gin and lemon juice

We can also request to mix ingredients and pretty much anything related to Baxter doing cocktails and above all also sequence of orders in a single request.

If we request something that is not on the recipes list or for which Baxter has not the ingredients (like a whiskey or a white russian) it will let us know with a pose as clear as possible.



OpenAI GPT-3

It is not the purpose of this presentation nor of this project to enter into the details of GPT-3 working structure and architecture. But here I can give some short glimpses of the working methodology.

The reason because this model is so suited for my task is that the network, which is really big (many billions of parameters, but it depends on the version of the implementation), is already trained. GPT stands in fact for Generative Pre-trained Transformer.

In practice this model can adapt to different tasks proposed by the end user thanks to a prompt, that is called context, that the same user will feed to it. This prompt will then be tokenized (i.e. split into tokens, short pieces of words that are long no more then 4-5 characters). It is also necessary to find the correct implementation of GPT for the task that is relevant to us. GPT is in fact used for all sort of modality, from image generation (DALLE 2) to code generation (Codex). These are tasks completely different from mine but that share the same NLP understanding model.

In my case scenario I’ve used the model as a sort of text completion. Programming it to return me the ingredients or multiple lists of ingredients to pick and pour with Baxter.

Part of the prompt used

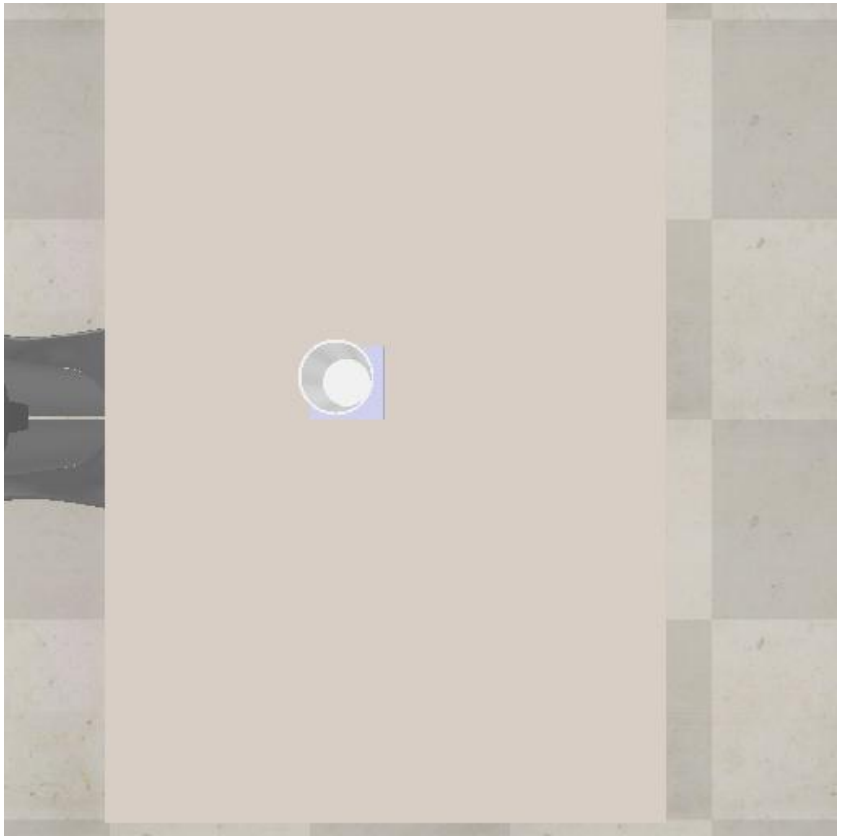
Baxter is a bartender robot that have to listen to orders of cocktails and it must return the ingredients. The orders can be of complex cocktails following a given recipe, or they can be of separate single ingredients. It is also possible for Baxter to receive a sequence of orders. If the order is not on the recipes list or it is not one of the four basic ingredients (Gin, Vermut, Campari, Lemon), Baxter will return "[?]". It is also possible to ask Baxter to give again an order already given, for example the last one.
[...]

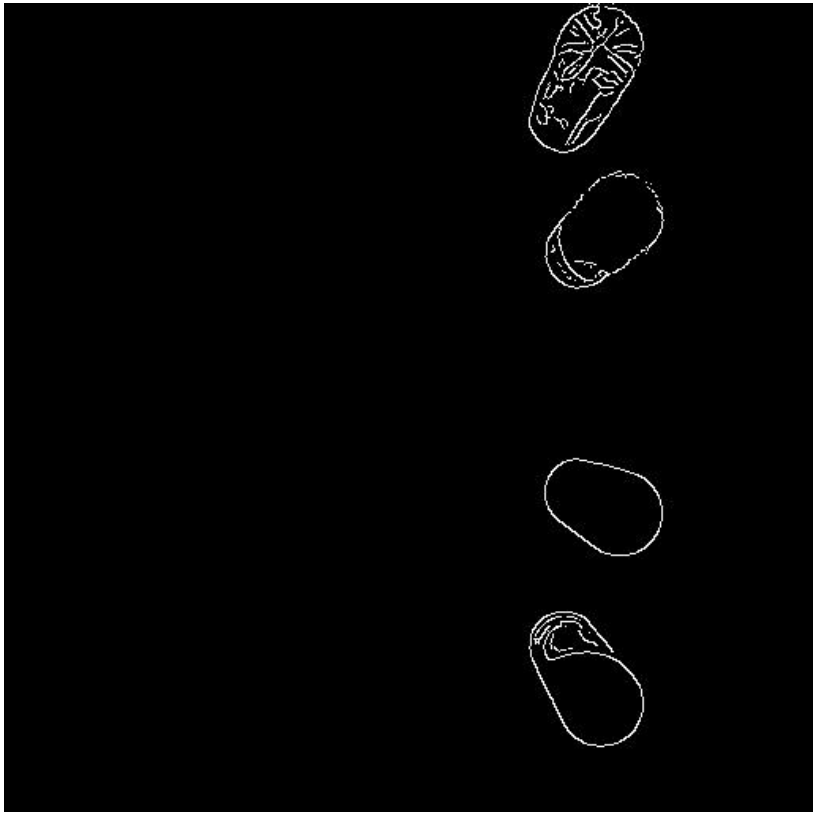
Results and conclusions

For the results I want to say that all the possible orders and requests already cited in the previous parts of this presentation work fine. Another possibility not yet mentioned is the capability of the model to show “memory”. To mimic phrases like “the same” that can be heard in a bar, it is possible to have the same order as the last one. In other words if one ask to Baxter a cocktail and then use a phrase like “the same” or another one equivalent in meaning, it will return the same cocktail as before.

To conclude, I think that this project has shown to me how a field like artificial intelligence can well embrace robotics, and above all how in the near future these will probably be fields that will get closer and closer.

An expansion…





Some references

- OpenAI, Language Models are Few-Shot Learners, 2020
- Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, Improving Language Understanding by Generative Pre-Training, 2018
- https://www.coppeliarobotics.com/helpFiles/
- https://github.com/stepjam/PyRep
- https://pyrep.readthedocs.io/en/latest/

Thanks for the attention!