

Zoomato Chatbot

André Gouveia e Joel Pires

University of Coimbra

`gouveia@student.dei.uc.pt` and `jfpires@student.dei.uc.pt`

Abstract

Within the scope of the curricular unit of Artificial Intelligence of the University of Coimbra taught by the teacher Hugo Gonalo Oliveira, a goal oriented conversational agent was developed that was able to interact through text with the user. Called 'Zomato Chatbot' and having a retrieval based model, it is able to hold trivial conversations with the user, although its main purpose is to provide details about restaurants in the Lisbon and Porto area based on API calls from the application of Zomato.

1 Introduction

This document is intended to describe the project that we did in the curricular unit of Artificial Intelligence of the University of Coimbra taught by the teacher Hugo Gonalo Oliveira.

1.1 Problem

The idea of the project was not defined initially. What was requested was to build a project that applies concepts of Artificial Intelligence. Accordingly to that, various themes were suggested and we discussed with the professor what would be a good project for us.

1.2 Motivation

We were motivated to build something really useful in real life and since the beginning we were enthusiastic in building some type of virtual assistant that would facilitate in some way the interaction between the user and some application and could satisfy the user request in a much faster and natural way.

So, we easily interiorized the idea that a conversational agent would be a good project's theme for us. Actually, there are many platforms and applications that rely on chatbots to deal with customer service. We wanted to do a chatbot which did not yet exist but at the same time that could help users interact with a popular application. The idea of doing a chatbot for Zomato Application came naturally then.

1.3 Goals

We idealized a chatbot capable of informing the users of Zomato services about a restaurant's name, its location, its contact, its daily menu, its score and its price. This would be really useful because the user wouldn't need to search in Zomato API but he simply has to chat with our bot in a very natural way and just go really straight to the point of what he/she really wants to know.

2 State-of-the-Art Review

We had the challenge of building an application in where we could use techniques of Artificial Intelligence. Various themes were suggested in order to select one and try to implement it. We decided to develop a conversational agent, but we had no idea on how to build one, the architecture that we should use and, more worrisome, a practical idea where we could apply it.

2.1 Literature Review Methodology

So we appropriately decided to search for academic papers and tried to investigate as many as possible in the academic community in order to find articles related to conversational agents. We ended up finding this one that it turned out being very useful for us:

O'Shea J., Bandar Z., Crockett K. (2011) Systems Engineering and Conversational Agents. In: Tolk A., Jain L.C. (eds) Intelligence-Based Systems Engineering. Intelligent Systems Reference Library, vol 10. Springer, Berlin, Heidelberg

Since we thought it was very complete and detailed, we decided to review only this paper because it was the one that most elucidated us and allowed us to have a general notion of what we had to do.

2.2 Exploring Different Types of CA's

The article clarified us that there are many types of conversational agents in addition to Chatterbots - these were actually the only ones that crossed our minds when we faced the challenge of building a CA. So, in fact, we can build:

- Spoken Dialogue System (SDS)
- Chatterbot
- NLP-based Dialogue Management System
- Goal Oriented CA
- Embodied CA

A SDS is definitely what we are not looking for, because a SDS is appropriate to convert speech into text and we were determined to do a chat-style conversational agent.

From the article we understood that the main goal of the Chatterbot is to prolong a conversation with a human as much as possible, without being worried to solve any particular problem, just chatting with him. We thought that it would be interesting to have a social component in our project, but it would not be our main focus because of two reasons:

- We needed a innovative thing, there are tons of Chatterbots out there
- We would need to build a good one, by other words, we would need to develop a chatbot that can really confuse the user if he's really talking with a bot or a human. That would be very complex and likely unachievable within the project's deadlines.

However, we should be able to maintain social conversation in a very limited way and, for that purpose, we read in the article that it would be required a script "that define the conversation to be executed by a pattern-matching engine. The scripts contain rules which in turn contain patterns. "[1] And the only thing that chatterbot has to do is answer to the user accordingly to the best match to one of its patterns.

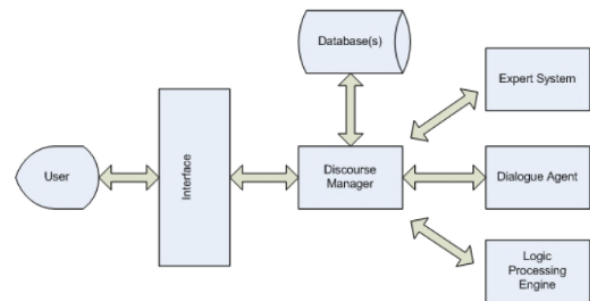
2.3 Ideas Began to Show up

When we analyzed the part that talks about Goal Oriented CA's we began to realize what would be a good and practical idea for our conversational agent. As we were going through that part of the article we understood that this type of CA has mechanisms that focus the conversation in order to achieve a goal, to solve a problem that the user needs. Without ignoring a natural language dialogue, it really goes straightforward in solving a user need, in giving as much information as possible about a business, an institution, a store, etc.

At this point we began to wondering what possible famous businesses still do not have actually some type of virtual assistant that can attend the doubts of its users. And finally we came up with the idea of a Zomato Chatbot. We already used the Zomato application and we realized how much easier could it be if instead of searching across the app for a good recommendation, we could directly talk with a chatbot that could easily recommend us a restaurant or give us any information that we want about it (location, menu, contact, etc.).

2.4 Implementation and Architecture

It would be difficult to implement this type of Goal Oriented CA? As the article says, "a pattern matching dialogue front end is combined with a rule-based system, which contains a model of the problem domain that is expressed in terms of a set of attributes." [2] Through the process of dialogue we have to capture attributes that can able us to identify the user's problem. The article also elucidated us about what should be a good architecture for a CA of that kind through this figure that was extracted from the article:



We would need a interface despite not being really important in this project. We also would need a database to long-term storage of user attributes that holds the answers of our conversational agent. Our system of rules would be our Expert System model. The logic processing engine could provide specific computational tasks. For example, in this case, this logic engine would be responsible to make the necessary API Calls to the Zomato API in order to retrieve the right answers to the user's needs.

2.5 The Importance of the NLP

We read that our Dialogue Management System can be based on Natural Language Processing (NLP). There are NLP procedures that are useful in our project like parsing, keyword extraction and formation of a structured lexicon. We realized that this is really important for what we want to do.

For example: the user wants to know the name of a restaurant in a certain location. How would we know the exact name of the location? We have to extract keywords like "eat", "lunch" and "dinner, and others in order to know the type of request that the user did (in this case he wants to have a meal somewhere and not, for instance, know the contact of a restaurant). But we have as well to know the semantic meaning of the words. We have to tag them with their semantic meaning: is the word a proper noun? Is it an adverb? Is it a preposition? So, to know the exact location we have to find the word 'no'/'em'/'na' and begin to extract the words after it. Every proper noun and proposition 'de'/'da'/'do'/'das'/'dos' that follows it will be considered part of the location's name.

Doing all this research and having a more detailed idea of what we have to do, becomes now much easier to implement it.

3 System Architecture

The first step in order to build a chatbot is to modelling one. We resolve then to see some academic articles that might could help us in this case and we ended up by seeing this useful one:

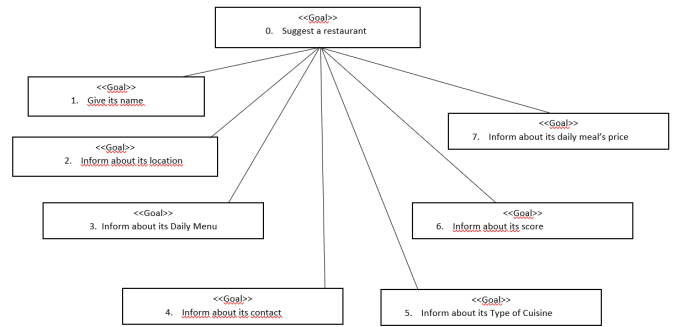
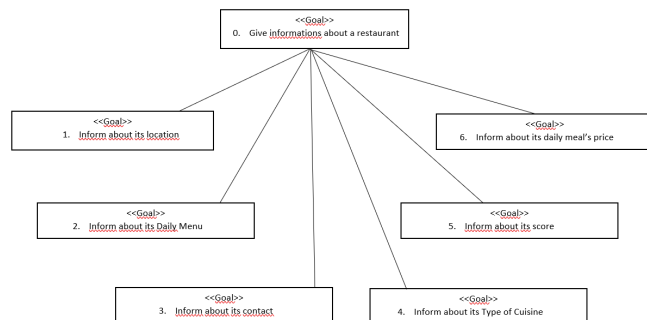
Scott A. DeLoach. Developing a multiagent conference management system using the o-mase process framework. In Agent-Oriented Software Engineering VIII, 8th International Workshop, AOSE 2007, Honolulu, HI, USA, May 14, 2007, Revised Selected Papers, AOSE 2007, pages 168–181. Springer, 2007.

One thing we knew so far: the interface won't be definitely our main focus. A user-friendly interface would be nice to implement but not an high priority. So in the beginning we chose to implement the chatbot on the terminal and only after that we should think if we can build a web app or put it in a platform like facebook.

3.1 Goal Model

As we decided that we were going to make a goal oriented conversational agent, the first thing that we had to search was about the goal models subtopics in the article. As it's said in it, "the first step in our O-MaSE compliant process is to create an initial Goal Model that captures the essential requirements of the CMS system as defined in the system definition or requirements documents." [3]

So we had to think: what are the main goals of our chatbot; what type of requests and needs will our chatbot satisfy? So, we decided to explore the Zomato API and tried to explore all kinds of information that it provides to the developer. Initially we purposed to build a chatbot that would inform the users almost anything they want as long that the information needed is provided by the Zomato's API. But, how could we design the goal model? We tried to follow the rules written in the article as well as its illustrations examples and we ended up with these following two schemes:



So we can divide the main goals of our chatbot into two:

- Give informations about a restaurant
- Suggest a restaurant

These two can be subdivided into similar sub-goals, like:

- Give the Restaurant's Name
- Give the Restaurant's Location
- Give the Restaurant's Contact
- Give the Restaurant's Daily Menu
- Give the Restaurant's Type of Cuisine
- Give the Restaurant's Score
- Give the Restaurant's Price of the daily meal

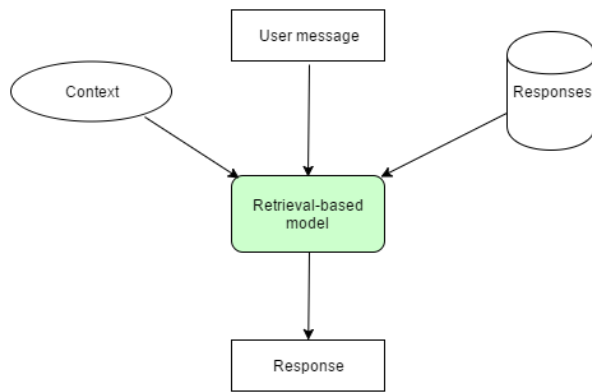
3.2 Retrieve Based Architecture

Initially we were undecided between two types of architectures:

- Generative Model
- Retrieval-Based Model

From our perspective, generative models are harder to build and train. Usually it requires a lot of examples (to do the training in a deep learning method). Also you can't be 100 percent sure what responses the chatbot will generate.

So we begin by implementing the retrieval-based model. Retrieval-based models are easier to implement and return more predictable results. It's expected that we won't obtain one hundred percent accuracy of responses, but one thing we know for sure - we know all the possible responses and can make sure that there are no inappropriate or grammatically incorrect responses. Besides that, these models are way more practical at the moment because many algorithms and APIs are readily available for developers. A scheme of the model is shown above [4]:



Initially we thought that our conversational agent should use the message and the context of conversation with the user for selecting the best response from a predefined statements list. In this case, the context could include the current position in the dialog tree or all the previous messages in the history of the conversation.

Despite of the fact that in the begining we planned to use the context, we concluded that would be impossible to do it within the project deadlines. Although, if the bot doesn't use context then it is stateless. It will only respond to the latest user message, disregarding all the history of the conversation.

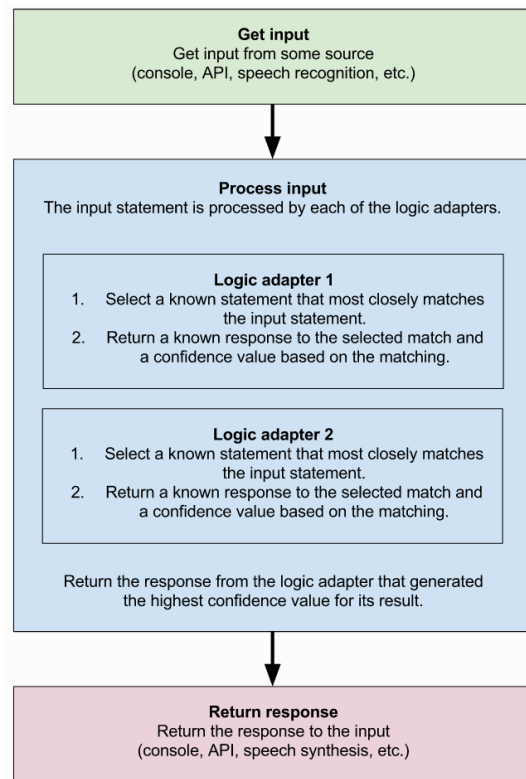
3.3 Tools

Chatterbot

In order to build our chatbot we used the Chatterbot[5]. ChatterBot is a tool that uses machine-learning to power its conversational dialog and is also built in Python. So, in fact it generates responses based on collections of known conversations. One interesting particularity of this tool is that the language design is independent which allows it to be trained to speak any language (including the portuguese!)

An untrained instance of ChatterBot starts off with no knowledge of how to communicate. However, each time the user enters a sentence, that sentence is saved in a library as well as the text that the statement was in response to.

As ChatterBot receives more input the number of responses that it can reply and the accuracy of each response in relation to the input statement increase. The program selects the closest matching response by searching for the closest matching known statement that matches the input, it then returns the most likely response to that statement based on how frequently each response is issued by the people the bot communicates with. The flow model is shown above [6]:



4 Performed Experiments

Portuguese Corpus

As we are portuguese, we think that a portuguese chatbot would be more appropriate and more familiar to talk with. To do that first of all we had to train it. Chatterbot already has a corpus data with portuguese words so we imported it and trained our agent.

After we trained it we tested it and we found out that the portuguese is not very good, it's quite limited in fact, but it gets better during long conversations (and it's expected taking into account that is a machine learning based conversational agent). Let's see a simple example of what we are talking about:

```

oi
Como vai você?
está tudo bem e com você?
Eu também estou.
ainda bem
obrigada
qual é o seu nome?
Eu poderia pedir uma xícara de açúcar?
sim
Posso ajudá-lo com alguma coisa?
para já não, obrigado
qual é o seu nome
Joel, e o seu?
Curso de Bacharelado em humanidades
  
```

(in white is the chatbot and in green is the user)

As we can see here, in a basic conversation we can say that the chatbot answers well just for the first 4 questions, from there on the chatbot doesn't make sense.

Zomato API

Obviously to build a chatbot that manages requests about Zomato information we had to deal with Zomato API. Fortunately, Zomato API is well documented although it's not very complete. It should be noted that it has some serious limitations.

For example, the API only returns, at max, 100 restaurants besides having thousands and thousands of them. This fact is responsible for sometimes the chatbot not being able to retrieve a restaurant name with some specifications written by the user despite of the fact of that actually exist restaurant that matches that specifications in the Zomato Application.

4.1 Pre-Processor

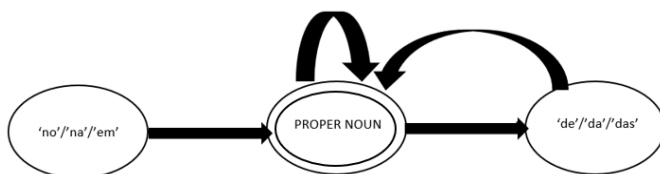
ChatterBot's preprocessors are simple functions that modify the input statement that a chat bot receives before the statement gets processed by the logic adapter. We build three preprocessors in the file *preprocessors.py* that:

- Clean excessive whitespaces inserted in the sentences
- Convert to ascii in order to recognize accented words and other special characters
- Convert every keyword that is presumably going to be used in a Zomato API Call into a valid keyword that actually can be used in that API. For example, if the user says "restaurante italiano", the chatbot converts the type of cuisine "italiano" to "Italian".

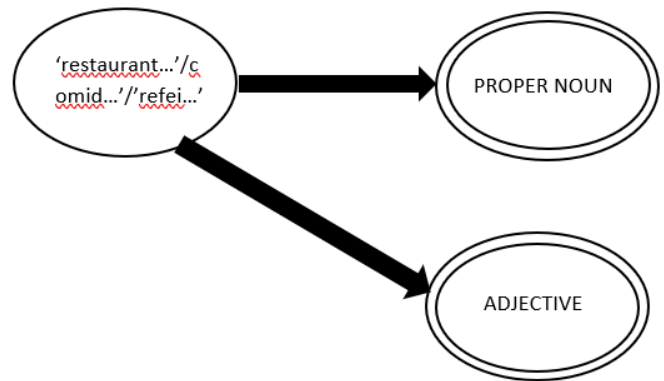
4.2 Logic Adapter

Logic adapters determine the logic for how ChatterBot selects a response to a given input statement.

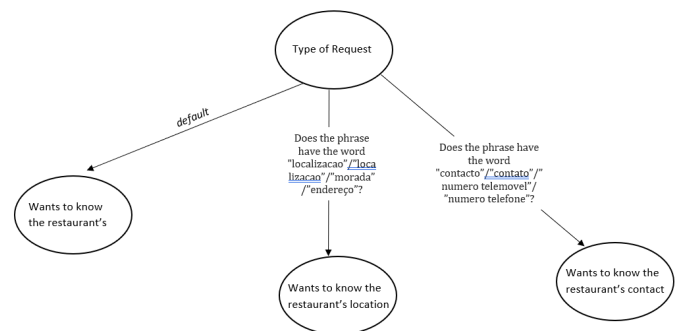
We really needed to understand the syntax and the semantic meaning of the sentences. We used taggers of the portuguese language (*tagger.pkl*) in order to tag each word of the sentence introduced by the user with the proper semantic value (pronoun, noun, adjective, etc.). Analyzing words and their semantic meanings, we can, for example, extract the requested location guiding us through this automaton:



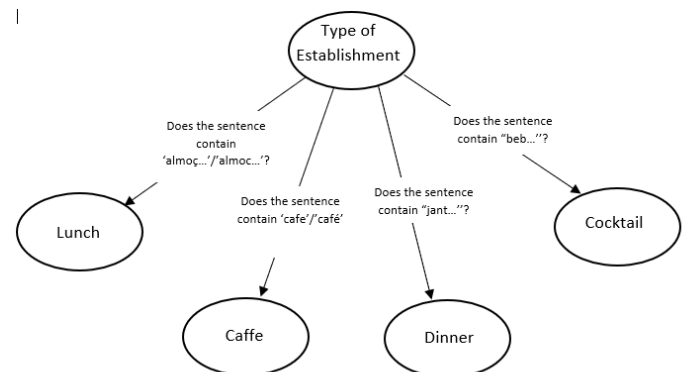
We can as well extract the requested type of cuisine guiding us through this automaton:



What about the type of the requested? We implemented in the source code this tree:



And if we want to know the type of establishment decide based on this tree:



We did some experiments by talking a lot with the robot. The following example shows us the final state of the chatbot:

```

01
Como vai você?
está tudo bem e com você?
Eu também estou.
és um robo?
Sim eu sou.
Olha, já agora diz-me um sítio para jantar no Chiado com menos de 10 euros
Boa-Bao é uma boa escolha.
Podes me dizer por favor qual o sítio mais barato para almoçar no Chiado que tenha um score mínimo de 3 estrelas?
A Padaria Portuguesa - Camões é uma boa escolha.
Diz-me um sítio para jantar no Chiado com menos de 10 euros
The B Temple é uma boa escolha.
Qual o restaurante italiano no Bairro Alto com o preço mais baixo?
NYC Slice Pizza é uma boa escolha.
Qual o melhor restaurante para comer no Chiado com um score superior a quatro?
Boa-Bao é uma boa escolha.
Podes me dizer por favor qual o sítio para almoçar no Chiado que tenha um score mínimo de 4 estrelas?
Boa-Bao é uma boa escolha.
Qual o restaurante na Saldanha com o score mais alto?
Qual o restaurante Italian no Bairro Alto com o preço mais baixo
Diz-me a localização do Boa-Bao
diz-me a localização do Mini-Bar
quero tomar café na Saldanha inferior a 10 euros e com no mínimo 4 estrelas
Fábrica dos Sabores é uma boa escolha.
Diz-me a localização do Choupana Caffé
diz-me a localização do Mini-Bar
qual a localização da República dos Cachorros?
Fica em: Praça dos Poveiros, 30, Santo Ildefonso, Porto

```

(in white is the chatbot and in green is the user)

As we can see, the chatbot randomly suggest a restaurant name and it is good at giving the location. It's also capable of understanding basic portuguese like asking 'How are you?' and other basics questions in portuguese. However, it is noticeable that it's not 100 percent accurate as we can see in the 4th and 2th answer counting from the last.

5 Conclusions

We think that the next step is to gift our chatbot with the hability to understande the context. It would be also very interesting to ask questions like "Podes me dizer um restaurante perto de mim?" that implies the chatbot to know the exact geolocation of the user. The natural language processing needs too to be improved. Besides that, we are very satisfied with the work done and most importantly we endorsed our skills in Artificial Intelligence methodologies and technologies.

6 References

[1], [2] O'Shea J., Bandar Z., Crockett K. (2011) Systems Engineering and Conversational Agents. In: Tolk A., Jain L.C. (eds) Intelligence-Based Systems Engineering. Intelligent Systems Reference Library, vol 10. Springer, Berlin, Heidelberg

[3] Scott A. DeLoach. Developing a multiagent conference management system using the o-mase process framework. In Agent-Oriented Software Engineering VIII, 8th International Workshop, AOSE 2007, Honolulu, HI, USA, May 14, 2007, Revised Selected Papers, AOSE 2007, pages 168–181. Springer, 2007.

[4] https://cdn-images-1.medium.com/max/1600/1*449TonHoHEvc-bwdXPr5OA.png - Image's URL

[5] <https://github.com/gunthercox/ChatterBot> - Chatterbot's Source Code

[6] <https://chatterbot.readthedocs.io/en/stable/> - Image's URL