

Programming a Chatbot in Python using Emotional Cognitive Conversational Agent Architecture (ECCAA)

Gnaneswari Gnanaguru

Jain University, New Horizon College, Bangalore, India

Corresponding author: contact_gnaneswari@yahoo.com

DOI: <https://doi.org/10.26438/ijcse/v7i3.510516> | Available online at: www.ijcseonline.org

Accepted: 06/Mar/2019, Published: 31/Mar/2019

Abstract— This research is an attempt to substitute an agent in place of a customer care executive in a helpdesk kind of scenario by building an emotional cognitive conversational agent. Some of the cognition that may be required to achieve human like dialog system are goal, desire, belief, intention, decision making, emotions, compassion, emotional pragmatics etc. The functionality of this Cognitive Architectures is Decision making, Prediction, Problem solving, Reasoning and Belief, action, communication between agents, Learning and reflection.

Conversational agent or Chatbot is become to rule our day to day lives. Alexa, Siri, Watson, etc have become the buzz words of today. This research paper is an attempt to implement the architecture for Conversational agents based on human cognition using Python and Machine Learning.

Keywords— Artificial Intelligence, Cognitive science, Chatbot, Machine Learning, Python, Conversational Agents

I. INTRODUCTION

Dr. Langley [1] says agents acquires knowledge about the environment through perception, implication of situations through reasoning and perception, knowledge is gained through learning, knowledge is also gained from other agent's experience/learning through communication. Cognitive architectures can be designed by clearly defining the functionality, provide a mechanism based on the functionality and implement the model in real time scenario. SMCA (Society of Mind approach to Cognitive Architecture) makes use of a generic architecture, and developed in terms of generic cognitive and metacognitive agent types. The proposed architecture is based on SMCA. [2]

II. COGNITIVE AGENT ARCHITECTURES

EM-ONE architecture proposed by P. Singh originated from Marvin Minsky's "emotion machine" architecture [3]. EM-ONE architecture for common-sense computing, that is

capable of reflective reasoning about situations involving physical, social, and mental dimensions. SMCA: (Society of Mind approach to Cognitive Architecture) was developed in terms of generic cognitive and metacognitive agent types. Each agent is designed to fit one of the following categories: reflexive agents, reactive agents, deliberative (BDI models) agents, learning, metacontrol and metacognitive agents .

The conversational agent architectures available were lacking the incorporation of the cognitive theory with them. My research is an attempt to incorporate the conversational agent architecture with the cognitive architecture so that the chatbot will sound more like a human.

III. EMOTIONAL COGNITIVE CONVERSATIONAL AGENT ARCHITECTURE (ECCAA):

The ECCAA is designed based on the hypothesis that 'There is an internal cognitive system in a conversational agent architecture that will map a syntactic structure to a semantic one based on pragmatics'.

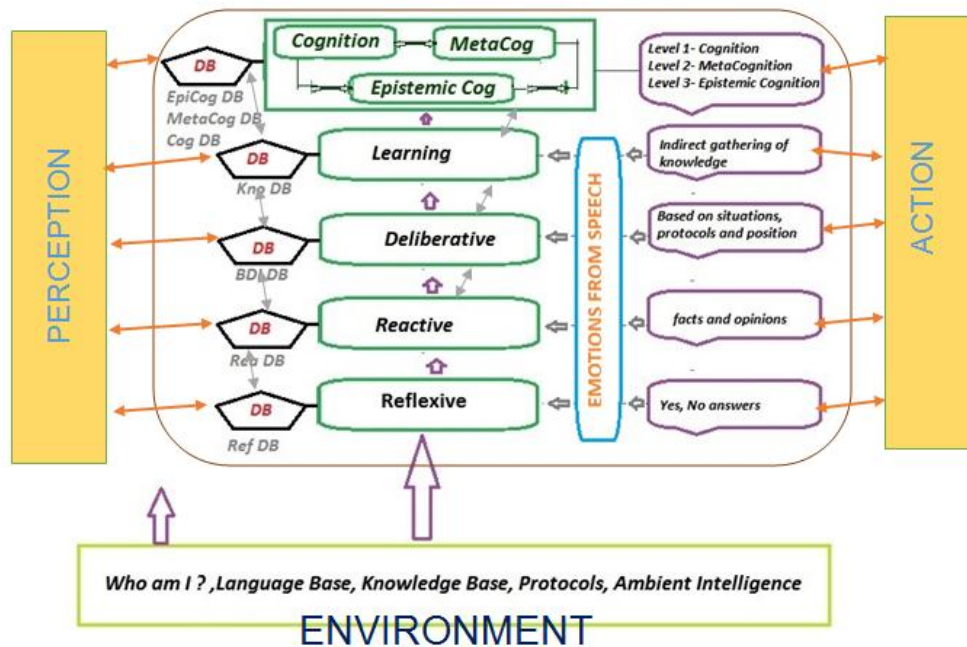


Fig.1 ECCAA

Emotional Cognitive Conversational Agent Architecture (ECCAA) is a 7 Layer architecture where each layer is connected to a specific database which stores successful conversations. Basically, the conversations are categorized into 5 different levels such as Reflexive layer, Reactive layer, Deliberative layer, learning layer and Cognition layer based on SMCA. But in the case of conversational agent architecture we need to further classify the Cognition layer. Thus, in ECCAA, the Cognitive layer is further classified into Cognition, Meta Cognition and Epistemic Cognition. Apart from the horizontal layers; there are vertical layers such as perception, action and emotion. [4] Thus ECCAA is language independent, flexible to all socio-cultural environments. The language and knowledge engine is used to store the syntax, semantics of the language and environmental, social facts respectively. Though the implementation is based on the English language, the model will be universal and will be robust irrespective of the language. [5]

IV. Implementation of the ECCAA Model

A simple Chatbot is programmed using Python and its machine learning packages.

A dataset is created based on the domain you want the Chatbot to build a knowledge base. Train the model using the above said architecture and test for accuracy.

4.1 Reflexive Layer:

In this layer only the yes/no responses are trained to the 'GGBot'.

Eg. Are apples red fruits? Is India democratic?

Here basic stimuli and responses can be trained from the knowledge base and meta data which is the information on the agent's involved in conversation are stored. The dataset used to train is 'ggbotyesno'.

Knowledge base of Domain includes Company policies, Do's and Don't's, Today's Rates, Trouble shooting queries, Clarification, Functioning and protocols, etc [6]

Meta Data is information about the Agent which includes Name, Address, Status, designation, family details, employer details, etc., Personality, state of mind, Status in the society, body language, facial expressions, emotions, etc., Desire, Belief, Intentions (BDI) and Goals.

#Emotional Conversational Agent Architecture using Cognitive Pragmatics

#ECCAAModel.py

```
from chatterbot import ChatBot

bot1 = ChatBot('GGBot1', trainer=
'chatterbot.trainers.ListTrainer')

bot1.trainer.export_for_training
('./export1.yml')

conversation=open
('ggyesno.csv','r').readlines()

bot1.train(conversation)

bot1.trainer.export_for_training
('./ggbotmodel1.csv')
```

Fig.2 Reflective Layer

4.2 Reactive Layer:

This layer is very similar to the previous layer, except that instead of on yes/no responses; responses are facts stored in the knowledge base and in the meta data. The dataset contains more stimuli-response with the responses derived from the facts stored in knowledge base together with the yes/no responses. The dataset used for this level of training is known as 'ggbotfacts'.

Eg. Who is the president of United States? What is your designation?

create the dialog file

```
conversation=open('ggbotfacts.csv','r').readlines()

bot2.train(conversation)

bot2.trainer.export_for_training('./ggbotmodel2.csv')
```

Fig.3 Reactive Layer

4.4 Learning Layer:

4.3 Deliberative Layer:

A Heuristic search is performed based on Neural Networks to choose the most accurate reply [7]. In order to train and perform heuristic search a large English corpus known as 'ChatterBotEnglishCorpus' is used along with the datasets 'ggbotyesno' and 'ggbotfacts'.

The training is as follows;

```
runfile('D:/PhD/phd2018
Implementation/GGBot/trainnew.py',
wdir='D:/PhD/phd2018
Implementation/GGBot')
GGBot Trainer: [#####]
100%
You: How was your day today?
Bot: Awesome.
0.85
You: What are you doing today?
Bot: Software Update is scheduled.
0.78
You: Good. Can I have some?
Bot: I am doing well, how about you?
0.58
You: I am fine. thankyou.
Bot: How are you?
0.86
You: I said.
Bot: Do you know all of it?
0.67
You: exit
```

Fig.4 Deliberative Layer

The numbers above indicate the percentage of accuracy which is also referred as confidence in the responses.

Print the responses

```
while True:
request = input('You: ')
if request == "exit":
break
i=i+1
response1 = ggbot.get_response(request)
threshlist1.append((response1.confidence)*100)
ilist1.append(i)
print('GGBot: ', response1)
print((response1.confidence)*100)
```

Fig. 5 Confidence of the Chatbot

In all the above models, learning is inevitable in a ChatBot. So practically learning is not a separate layer. Whenever there is a response for a stimuli, the new response is stored in the database. The databased use is SQLite3 database.

```
#Create and Connect to SQL database
bot = ChatBot('GGBot',
storage_adapter='chatterbot.storage.SQLiteStorage
Adapter',database='./datavase.sqlite3')
# It happens after the learning layer

ifget_feedback():

bot.learn_response(response, input_statement)

bot.storage.add_to_conversation(CONVERSATION_ID, statement, response)
```

Fig. 6 Learning Layer

Only successful conversations are stored in the database [8]. Inorder to filter conversations we have feedback system.

4.5 Cognition Layer:

The core of the ECCAA is the cognitive layer. 'Emotion' is added to the dataset. Before training the dataset, every stimuli and response is analyzed and classified based on the emotions. This is done using an emotion detector API known as 'indico'. It is capable of detecting which is the dominant emotion in the sentence. The basic emotions considered are anger, joy, fear, sadness, surprise. For every statement, an vector value is derived for every emotion. Now training is performed as usual with all the stimulus and responses with emotions. [9]

The emotion with highest vector value is assigned to that particular sentence.

```
D:/PhD/phd2018 Implementation/indigoemot.py',
r:/PhD/phd2018 Implementation')
': 0.1811702698469162, 'joy': 0.22648397088050842,
23824439942836761, 'sadness': 0.17388318479061127,
': 0.18021820485591888}, {'anger':
!544100761414, 'joy': 0.24268311262130737, 'fear':
!418488025665, 'sadness': 0.15005141496658325,
': 0.21510586142539978}, {'anger': 0.353613942861557,
': 0.31560152769088745, 'sadness': 0.144
17325592, 'fear': 0.13417458534240723, 'joy':
35872066021}, {'anger': 0.1492113471031189, 'joy':
!083523750305, 'fear': 0.2272227257490158, 'sadness':
289924383163, 'surprise': 0.06976218521595001},
: 0.353613942861557, 'surprise': 0.31560152769088745,
': 0.14459635317325592, 'fear': 0.13417458534240723,
!520135872066021}, {'anger': 0.19296512007713318, 'joy':
!36751794815, 'fear': 0.43771886825561523, 'sadness':
!721774816513, 'surprise': 0.08980514854192734},
: 0.23729276657104492, 'surprise': 0.3912753760814667,
': 0.2236287146806717, 'fear': 0.1064252182841301, 'joy':
!793555855751}, {'anger': 0.1692388951778412, 'joy':
!293561935425, 'fear': 0.1937219202518463, 'sadness':
!549790382385, 'surprise': 0.21972069144248962}}
```

Fig.7 Emotions in Cognitive Layer

4.6 MetaCognition Layer:

This layer is more intelligent than the above layers. This layer is self-reflective. This can be implemented in which the chatbot has to choose responses with confidence value more than the threshold assigned. If the confidence is say < 65% then, the response must be a specific response such as 'I am sorry, I don't know.'

```
In [10]: runfile('D:/PHD/phd2018 Implementation/GGBot/trainmeta.py', wdir='D:/PHD/
phd2018 Implementation/GGBot')
List Trainer: [#####] 100%

You: hi
Bot: Hello
1.0

You: Will there be another chance?
Bot: I am sorry, I do not know.
1

You: exit
```

Fig. 8 Metacognitive Layer

4.7 Epistemic Cognition Layer:

In Epistemic cognition level, the agent reflects on the limits of its knowledge and overcome such limitations. This is the highest level in ECCAA. It is not only ‘self-reflective’ but also learn by itself. This is implemented by adding responses directly to store in the database that is specified in the ‘Storage Adapter’. This added response is added to the trained dataset.

```
In [8]: runfile('D:/PhD/phd2018 Implementation/GGBot/trainepticog.py', wdir='D:/PhD/
phd2018 Implementation/GGBot')
List Trainer: [#####] 100%
You:
hello
How are you?

Is "How are you?" a coherent response to "hello"?

yes
You:
What is the name of this robot?
Barajas

Is "Barajas" a coherent response to "what is the name of this robot"?

no
please input the correct response to be added to my knowledge base

My name is GGBot developed by Gneswari.
Response added to GGbot!
```

Fig.9 Epistemic Cognition Layer

V. RESULTS

A questionnaire was prepared commonly for all the models in order to measure their performances. A graph was plotted with responses in the x axis and the confidence value in the y axis.

```
import matplotlib.pyplot as plt

plt.plot(illist1,threshlist1, color='green')

plt.xlabel('Responses')

plt.ylabel('Confidence Level')

plt.title('Confidence Level in responses in Model 1')

plt.show()
```

Fig. 10 Response Graph

The result was as follow for various models;

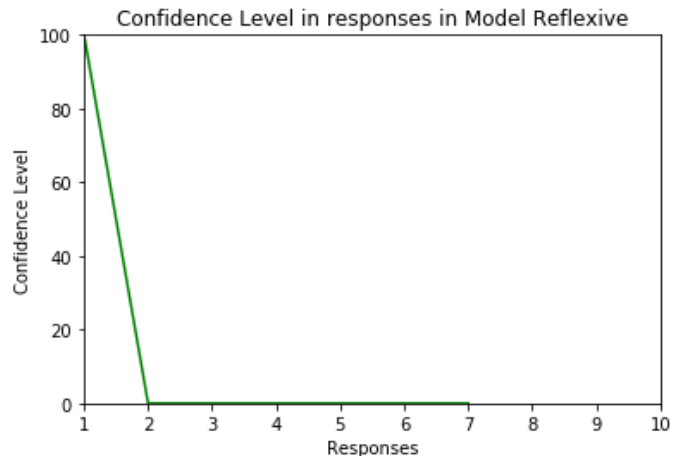


Fig. 11 Response Graph for model 1



Fig. 12 Response Graph for model 2

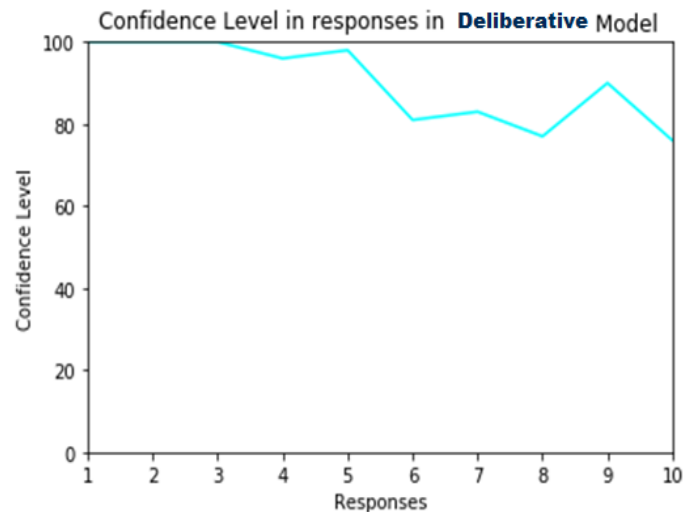


Fig. 13 Response Graph for model 3

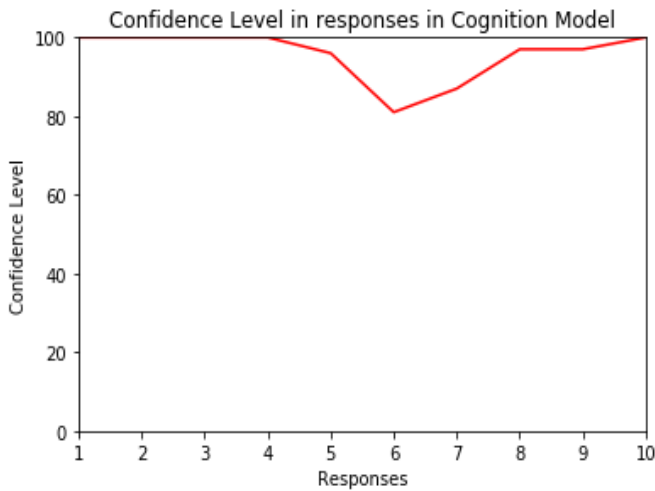


Fig. 14 Response Graph for model 5

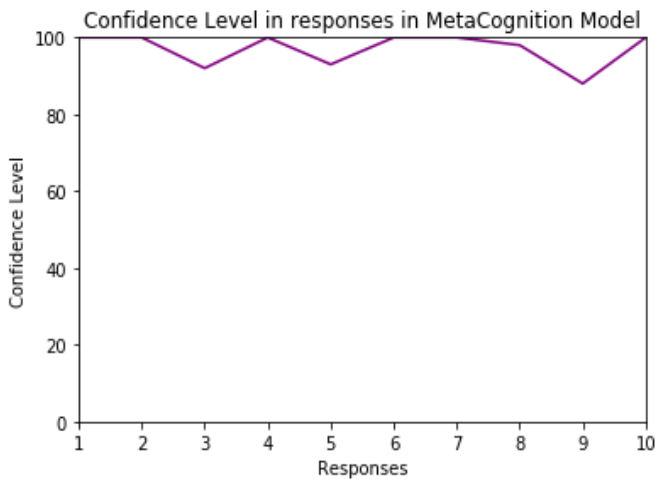


Fig. 15 Response Graph for model 6

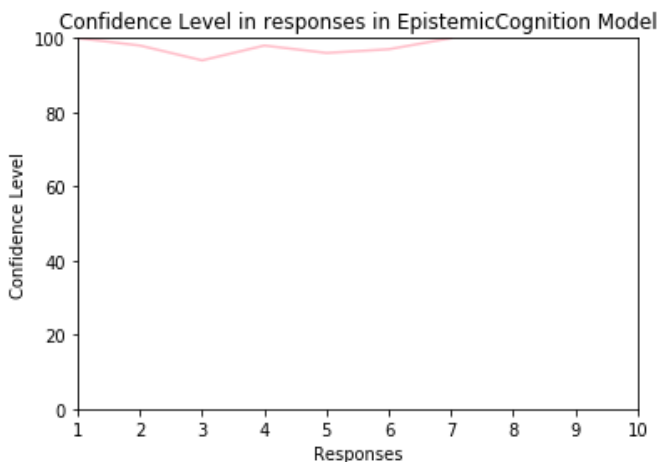


Fig. 16 Response Graph for model 7

VI. CONCLUSION

Thus, a conversational agent may produce accurate results, but in order to achieve results that are close to the human responses in conversations, the cognitive model ECCAA is suggested it is a very efficient one. This model can be incorporated to any conversational agent architecture. This model also works for training multilingual conversational agents with respective datasets.

REFERENCES

- [1] Pat Langley, John E. Laird, Seth Rogers, Cognitive architectures: Research issues and challenges, Cognitive Systems Research, ISSN 1389-0417, Volume 10, Issue 2, Pages 141-160, 2009
- [2] Venkatamuni, Vijayakumar Maragal. A Society of Mind Approach to Cognition and Metacognition in a Cognitive Architecture, Dissertation of Doctor of Philosophy in Computer Science and Engineering, University of Hull, London, 2008
- [3] Gnanaguru Gnaneswari, Venkatamuni, Vijayakumar Maragal, Building a Conversational Agent based on the principles of Cognitive Pragmatics using Cognitive Architecture, International Journal of Engineering Research & Technology (IJERT), ISSN: 2278-0181, Vol. 6 Issue 02, February 2017.
- [4] AbuShawar, Bayan & Atwell, Eric., ALICE chatbot: Trials and outputs, Computación y Sistemas, Vol. 19, No. 4, 2015, pp. 625-63, 2005
- [5] Arend Hintze, Understanding the four types of AI, from reactive robots to self-aware beings, Michigan State University, 2016.
- [6] P Singh, EMONE: An Architecture for Reflective Commonsense Thinking, Dissertation of Doctor of Philosophy in Computer Science and Engineering. Cambridge, MA: Massachusetts Institute of Technology, 2005.
- [7] Weitzenfeld, A., Arbib, M., Alexander, A.: NSL—Neural Simulation Language: A System for Brain Modeling, MIT Press, Cambridge, MA, 2002.
- [8] N, Davis D., Computational Architectures for Intelligence and Motivation. International Symposium on Intelligent Control, Vancouver, Canada :17th IEEE, 2002.
- [9] A.K.Gupta, S.Gupta, "Neural Network through Face Recognition", International Journal of Scientific Research in Computer Science and Engineering, Vol.6, Issue.2, pp.38-40, 2018

Authors Profile

G.Gnaneswari is a research scholar in Jain University. She is working as an Assistant Professor in the Department of Computer Applications at New Horizon College, Banagalore, India. Her area of interest includes Artificial Intelligence, in particular Cognitive Architectures, Machine Learning, Robotics etc., She is doing her research based on building an Cognitive Architecture for Conversational Agents.



Architecture for