

# -Answer My Question- Bot

By Erik di Biase & Filippo Libardi

## Introduction

Trying to show the very bottom lines of machine learning.

Where in each knowledge-based system, a given agent starts with basic grammar rules and empty databases.

It will simultaneously conversate and gain knowledge by what is told to him.

## Technical Details

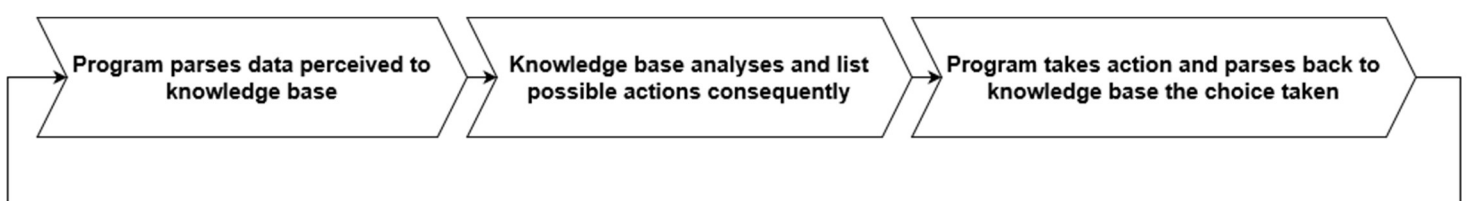
Written in:

- python 2.7

External Modules Imported:

- NLTK for NLP
- uuid4 for unique ID's generation
- sqlite3 for SQL databases management
- Python-telegram-bot is a python interface for the official Telegram bot API

Analysis & Thought Process:



## Research Topics

### Natural Language Processing:

Interpreting human text inputs, understanding the topic of the sentence, is something that can only be achieved by having a massive database of words and samples, this was provided to us by NLTK library. The Natural Language Toolkit, or more commonly NLTK, is a suite of libraries and programs for symbolic and statistical natural language processing for English written in the Python programming language.

Having a given sentence, the module provides some tools for dividing it in “clusters”, it then generates by a developer-decided algorithm a hierarchy diagram of all the words assigning them a probability of belonging to a certain topic based on the words next to it. It so generates a class-based language model, also known as cluster n-gram model.

### Finite State Machine:

The main conversation handler agent can change its behaviour in runtime, it in fact determines the state it is in. The states will be explained in the next class diagram. The agent sits in a deterministic environment, so each single choice is based on analysis of the sentence provided.

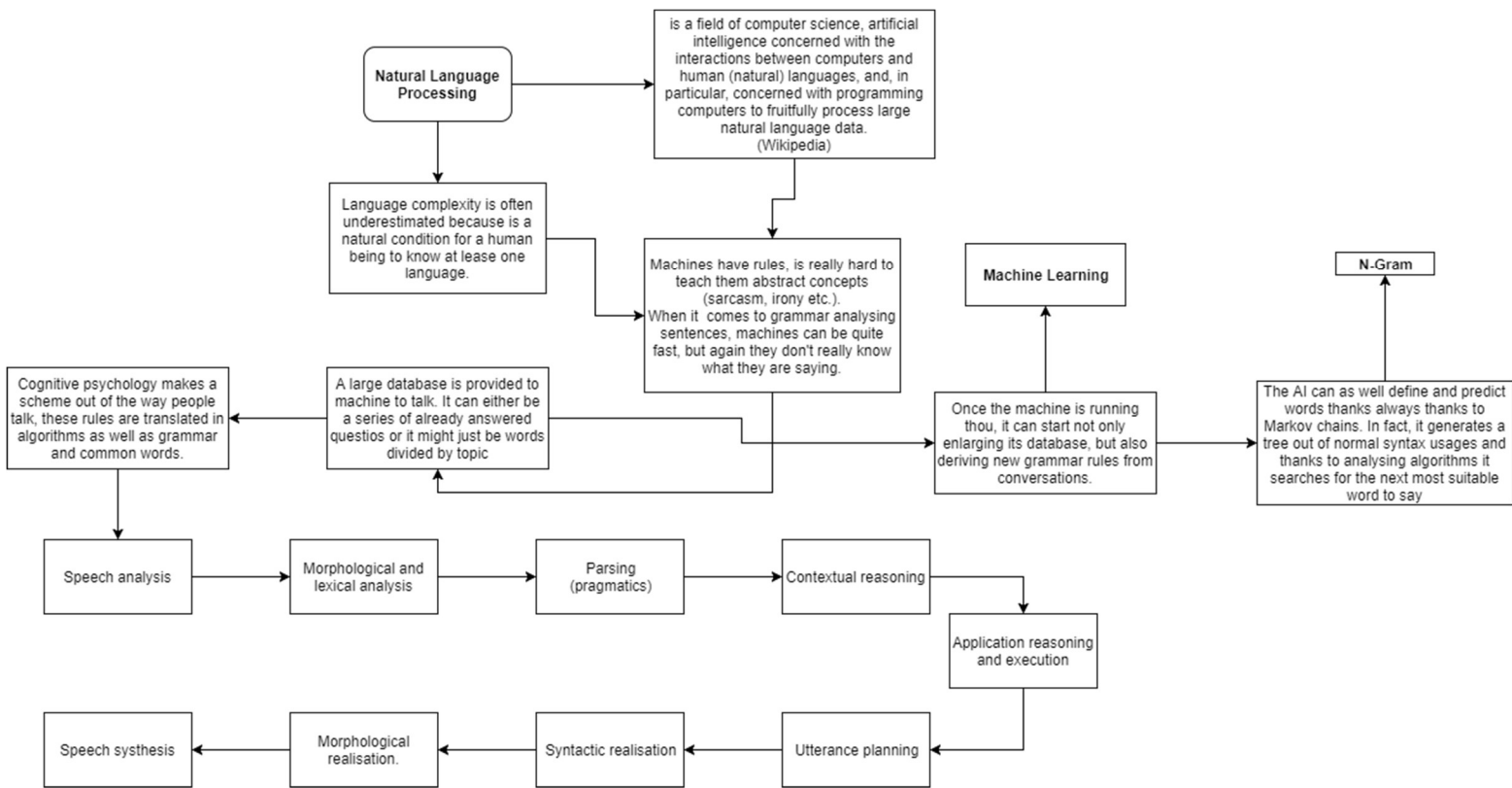
### Machine Learning:

Instead of providing the agent a large local database, we decided to give it only a small .db file where it will store user's details, questions corresponding answers (three tables in total).

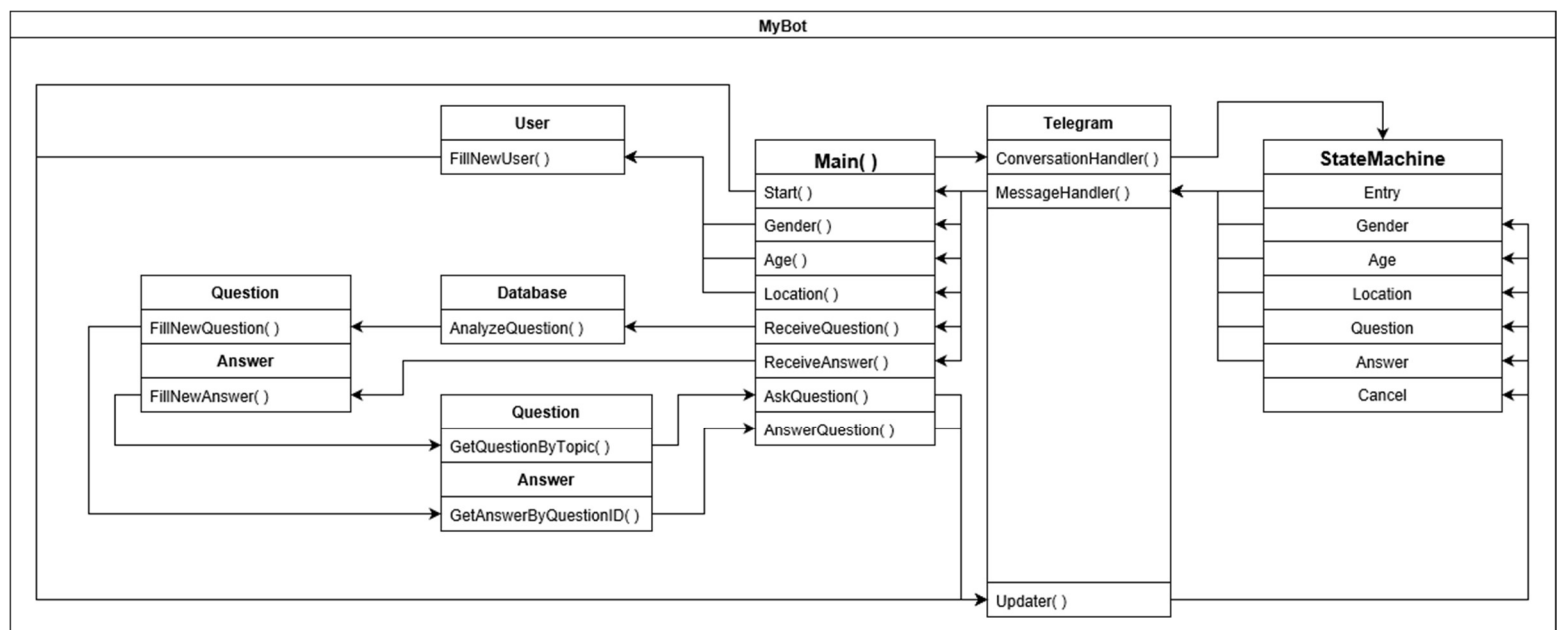
Only when it will need to solve language issues it will appeal to a module provided optimised database (see next point).

On any other occasion it will start not only enlarging its local databases but will also be in the condition to retrieve user's data of who answered what and returning statistics.

# Topics Diagram



# Class Diagram



## Conclusions

After all, we are satisfied with the project as a proof of our concept of machine learning, although it does not achieve all the tasks we thought of, there is a solid structure on the top of which we can implement late features, like generating statistics.

The whole process however, allowed us to extend our knowledge about A.I systems, Natural Language Processing and how to configure a chat-bot.

## Issues Assessment

Any development issue has been assessed as best as possible.

Issues Assessed	Solutions Applied
<ol style="list-style-type: none"><li>1. Telegram library unable to receive multiple message update during each state</li><li>2. The bot was unable to distinguish new and old questions, confusing the user</li><li>3. The manual implementation of grammatical rules and semantic analysis, could only have been completed with an unbelievable amount of time which wasn't available</li></ol>	<ol style="list-style-type: none"><li>1. The bot has been adapted to perform question analysis and answer during sub-state</li><li>2. A pre-emptive analysis of the current questions database status, allows the bot to switch topic or end the conversation when all question have been asked</li><li>3. The bot uses a simple implementation of the NLTK lexicon, which can perform the semantic analysis required without using much computational power</li></ol>