Conversation Bot in python

By Erik Di Biase & Filippo M. Libardi

Introduction

Trying to show the very bottom lines of machine learning.
Where in a given 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.

The Project

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

Procedure

1. Program parses data perceived to knowledge base

2. Knowledge base analyses and list possible actions consequently (is a question or an affirmation)

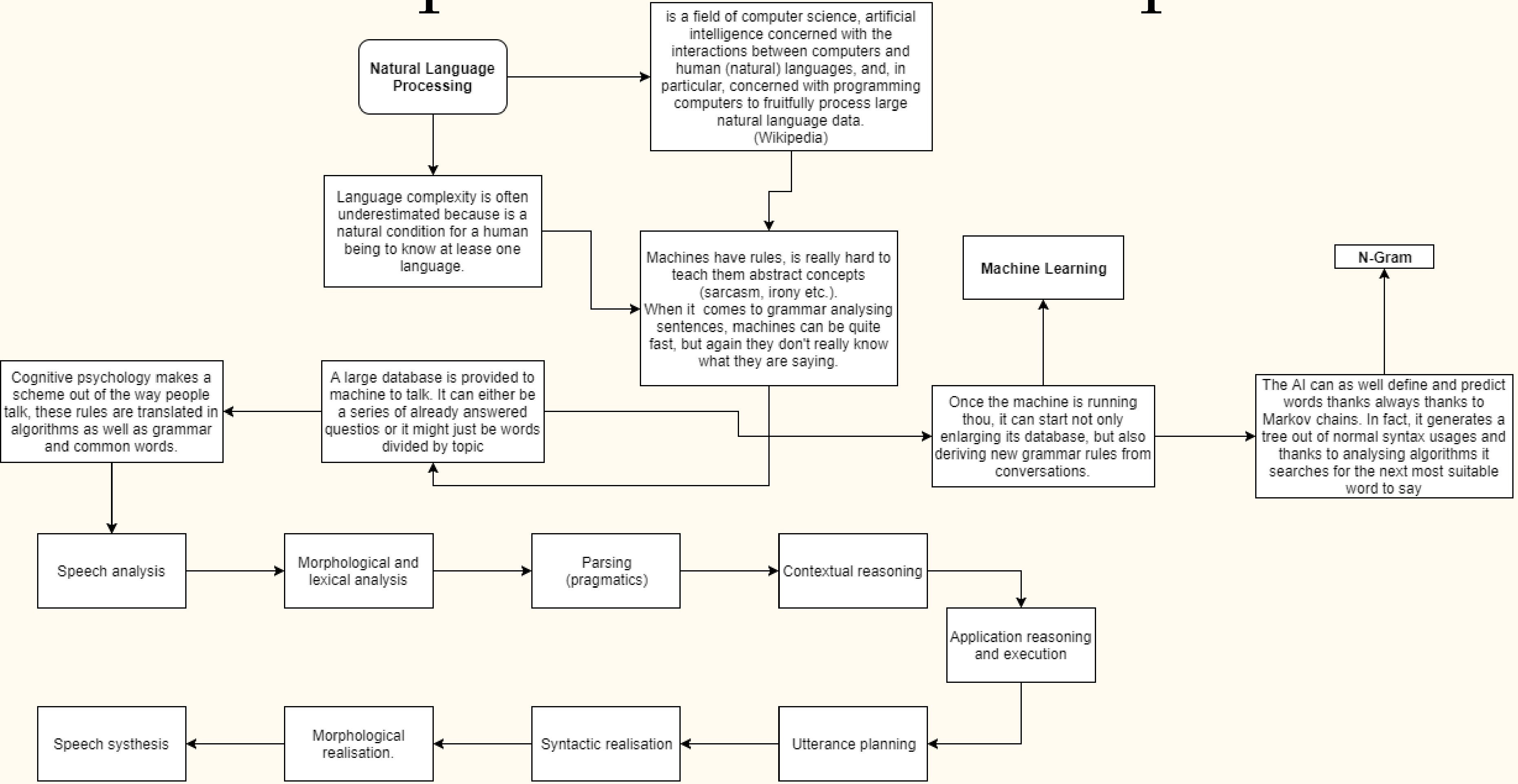
3. Program takes action and parses back to knowledge base the choice taken

Topics covered and researched:

Natural Language Processing: Interpreting human text inputs, understanding what is the topic of the sentence is something can only being achieved having a massive database of words and samples, this was provided 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*.

Topics covered concept:



is a field of computer science, artificial intelligence concerned with the interactions between computers and human (natural) languages, and, in particular, concerned with programming computers to fruitfully process large natural language data. (Wikipedia)

Language complexity is often underestimated because is a natural condition for a human being to know at lease one language.

Machines have rules, is really hard to teach them abstract concepts (sarcasm, irony etc.). When it comes to grammar analysing sentences, machines can be quite fast, but again they don't really know what they are saying.

A large database is provided to machine to talk. It can either be a series of already answered questios or it might just be words divided by topic

Cognitive psychology makes a scheme out of the way people talk, these rules are translated in algorithms as well as grammar and common words.

Once the machine is running thou, it can start not only enlarging its database, but also deriving new grammar rules from conversations.

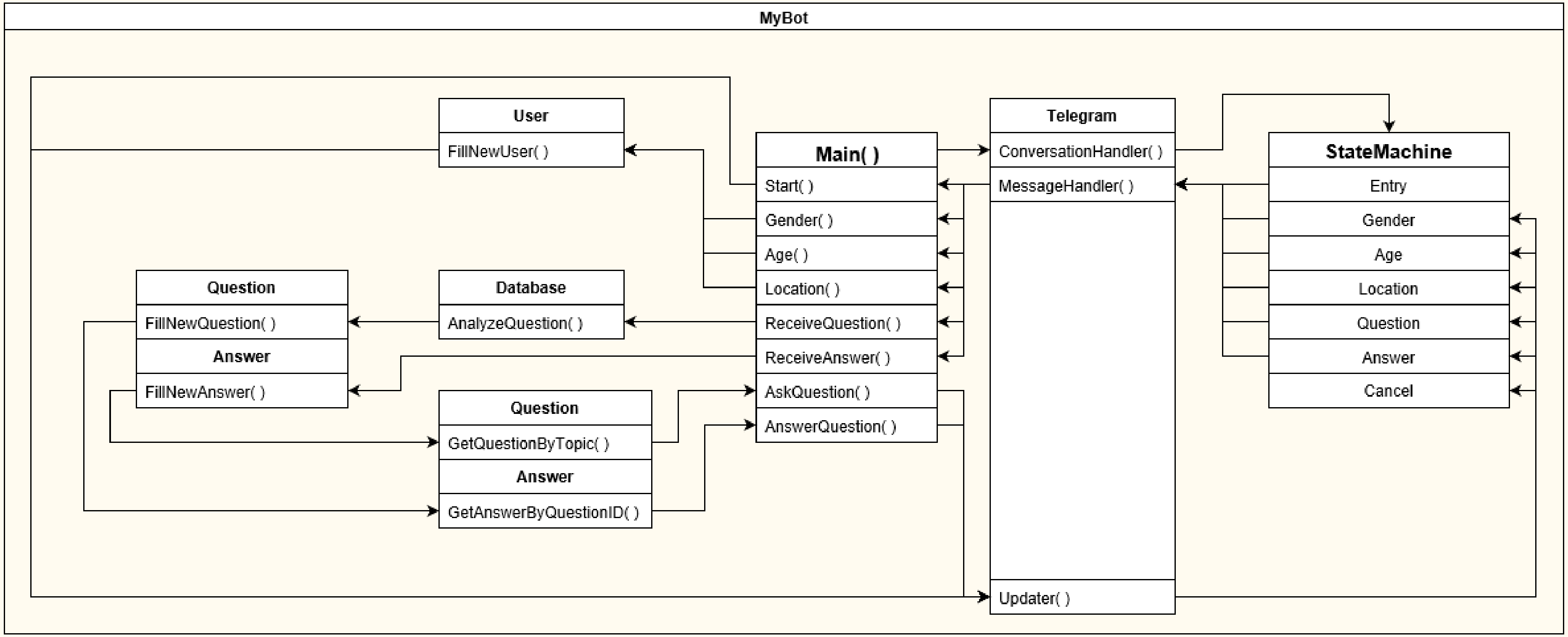
The AI can as well define and predict words thanks always thanks to Markov chains. In fact, it generates a tree out of normal syntax usages and thanks to analysing algorithms it searches for the next most suitable word to say

Topics covered and researched:

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 databases, we decided to give it only a small .db file where it will store users 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.

States Transition Diagram



Reflections

Issues assessed	Solutions applied
● Telegram library unable to receive multiple message update during each states	● The bot has been adapted to perform question analysis and answer during sub-state
● The bot was unable to distinguish new and old questions, confusing the user	● A preemptive analysis of the current questions database status, allows the bot to switch topic or end the conversation when all question have been asked
● The manual implementation of grammatical rules and semantic analysis, could only have been completed with an unbelievable amount of time which wasn’t available	● The bot uses a simple implementation of the NLTK lexicon, which can perform the semantic analysis required without using much computational power

Conclusions

Afterall, 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 (as it does not generate statistics), there is a solid structure on the top of which we can implement additional features.

The whole process however, allowed us to extend our knowledge about A.I systems, Natural Language Processing and how to configure a bot project.
Each issue has been assessed correctly and the we finished in time with the project plan.