MINISTRY OF EDUCATION OF REPUBLIC OF MOLDOVA

Technical University of Moldova

Faculty of Computers, Informatics and Microelectronics

Department of Software Engineering and Automatics

Report

on Artificial Intelligence Fundamentals

Laboratory Work nr. 1

Performed by: **Elizabet Grinciuc**, FAF-172

Verified by: **Mihail Gavrilița**, asist. univ.

Chișinău, 2021

Contents

Expert Systems

Finally, your dream came true and you've landed a job at "HeinleinAI" – the biggest AI company in the whole Luna-City! You're on a testing period so you will want to make sure that you do your best.

While you are daydreaming about free lunch, the mentor comes and hands you your first task – you'll need to build an expert system. They say that while going through the central library database they stumbled upon this ancient approach that looked like it could solve a particular lunar problem – detecting tourists.

The tourists on Luna-City are a big source of income and while almost every salesman and hotel administrator can easily tell one from a Loonie, our mentor researches a more systematic way of detecting them. While this would prove a trivial task to the city's main computer, resources are scarce these days and so you will be researching alternative, more special approaches.

For starters, research the types of tourists that visit Luna-City and collect a database of at least 5 tourist types and the criteria by which they can be distinguished from Loonies and between themselves (i.e. clothes, accent, gait, height and opinion on politics). After your database is done, create a system that would allow the user to answer some questions about a possible tourist. If the set of given answers matches a type of tourist from the database, this should be the system's answer. If on the other hand, the system determines that the person in question is a Loonie, the answer should be returned accordingly. Make sure to consider the case when the set of answers does not find a match in the database (highly improbable if you've done your research well).

The Task

To create an Expert System which will detect specific items based on characteristics of these items.

The system is based on user interaction, where the system asks and the user answers questions. Based on the answers, the Expert System will detect potential items which suit the characteristics from user answers.

The System should be clever and not ask useless questions (useless questions = questions, the answer of which won't affect the outcome).

Solution Description

The Expert System is a Moldavian one, that is why we decided to detect Moldavian poeple on the moon (hoping they will leave Earth soon). The System is constructed on functions, each of the function has one of the following roles:

1. Ask a random question, among those available
2. Filter the Moldavians and leave only those who meet the characteristics

3. Check if Final Result is ready
4. Check for useless questions (questions, the answer of which won't affect the outcome)
5. Check for remaining available questions to put
6. Print intermediate and final results

Code and Mentions

```python
import random

# let's detect Moldavians on the moon. They will surely be a good source of income for moon HoReCa

mold1 = ['russian', 'vodka', 'mamaliga', 'manele', 'Independence Day']
mold2 = ['romanian', 'wine', 'sarmale', 'rock', 'Wine Day']
mold3 = ['moldavian', 'wine', 'sarmale', 'manele', 'Wine Day']
mold4 = ['romanian', 'wine', 'ciorba', 'pop', 'Romanian Language Day']
mold5 = ['moldavian', 'wine', 'sarmale', 'rock', 'Independence Day']
mold6 = ['russian', 'vodka', 'ciorba', 'pop', 'Wine Day']
mold7 = ['russian', 'vodka', 'ciorba', 'pop', 'Wine Day']

# differencesList = ['mother language', 'favorite drink', 'favorite food', 'favorite music', 'favorite holiday']
moldavians = [mold1, mold2, mold3, mold4, mold5, mold6, mold7]
molds = moldavians

# here we declare list of differences/questions which can be put; and double it in another variable
diff = ['mother language', 'favorite drink', 'favorite food', 'favorite music', 'favorite holiday']
diffP = ['mother language', 'favorite drink', 'favorite food', 'favorite music', 'favorite holiday']

# here we stock the answers that user gave
answers = []

# here we manage the question putting and the removing of the question from the list of possible questions
def question(choice):
    print('What is their ' + choice + '?')
    answer = input()
    diffP.remove(choice)
    return answer

# here we filter moldavians based on the answer of user
def filterOptions(answer):
    global molds
    moldsToRemove = []
    for mold in molds:
        if answer not in mold:
            moldsToRemove.append(mold)
    for moldToRemove in moldsToRemove:
        molds.remove(moldToRemove)

# here we will check which questions are useless to ask, like if remaining molds have common differences (e.g. both like wine)
def checkForUselessQuestions():
    global diffP
    result = set(molds[0])
    for currSet in molds[1:]:
        result.intersection_update(currSet)
    result = list(result)
    if result:
        for item in result:
            index = molds[0].index(item)
            itemToRemove = diff[index]
            if itemToRemove in diffP:
                diffP.remove(itemToRemove)

# here we check if there are anymore questions to ask. if not, to print the final result
def checkForRemainingQuestions():
    if (len(diffP) == 0):
        printFinalResult()

# here we check if final answer is ready based on the remaining moldavians list
def isFinalAnswerReady():
    return len(molds) == 1

# we print intermediate/final results
def printIntermediateAnswer():
    print('INTERMEDIATE Answer: Potential moldavians are: ')
    print(molds)

def printFinalResult():
    print('FINAL Answer: The corresponding moldavian(s) based on your answers: ')
    print(molds)
    print('P.S. Your answers were: ')
    print(answers)
    exit()

# we start the poll
def startPoll():
    global molds
    global diffP
    global answers
    for i in range(len(diffP)):
        answer = question(random.choice(diffP))
        answers.append(answer)
        filterOptions(answer)
        if isFinalAnswerReady():
            printFinalResult()
        else:
            checkForUselessQuestions()
            checkForRemainingQuestions()
            printIntermediateAnswer()

startPoll()
```

Conclusions

The Expert System is based mostly on a filtering system and a system that puts correct questions. It was important to make the system to analyze well remaining questions to ask, after each round (each answer got from user input).

It was interesting to create such a system, because it may be used in mechanisms of potential clients detection for Google Analytics and Facebook adds. It was a nice experience. However, I'm interested in how this can be written clearer and which systems are used by big companies, where data that they analyzed is much bigger. So, the question - how to optimize the system and make it more performant and scalable?

References

[1] Elizabet Grinciuc. Source code for the laboratory work. Accessed February 13, 2021. https://github.com/ElizabetG/fia_lab.