## **Artificial Intelligence**

**Assignment 3**

## **Programming in Logic**

**NIM-BSCS-2020-04**

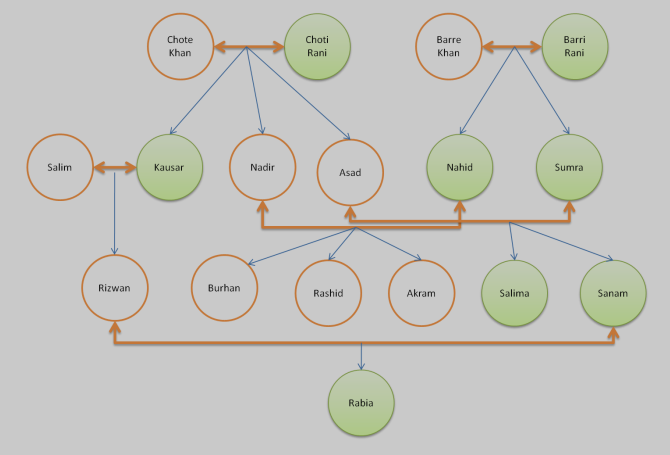
**Syed Muhammad Muneel Raza Naqvi**

**NIM-BSCS-2020-03**

**Shamsa Batool**

**Problem Understanding:**

This assignment is aimed at dealing with questions related to a family tree in human language. The family tree given in this assignment is a graphical representation of different individuals of the family linked with each other depending upon their relationships. First, we have to read that tree, and then ask questions in natural language which are supposed to be answered in accordance with the family tree. For example, you want to know who is the brother of Kausar. You may query in natural language and expect a response in natural language. So, the expected response would be, “Nadir and Asad are brothers of Kausar”.



**Proposed Solution:**

**Chatbot;To Bridge PROLOG and Python:**

To solve the given problem, we have to build a simple interface in Python through which our user can interact to give queries and get responses in return. For this purpose, first we have to translate the given relationships shown in the graphical tree into facts. These facts depict that two persons are related to each other in a specific way. For example, it is a fact that Chote Khan is the husband of Choti Rani, and Choti Rani is the wife of Chote Khan. So, we used a Programming Language, PROLOG which uses facts and rules to process and respond to queries. In PROLOG, facts are true statements about our data and rules are derivations of facts which are used to respond to queries about facts. In this processing, PROLOG simply checks the pattern of words stored in the facts and rules, and the query given by the user.

**Facts:**

We are provided with multiple facts about the Khan Family as Mianbiwi, Parents and Gins. These facts cover all the individuals, their husband-wife, parent-child relationship and gender. For example, husband-wife is represented as follows:

mianbiwi(Husband, Wife).

**Rules:**

In the facts, we were given two basic relationships and the gender of each individual. Now, to find different relationships among these individuals, we need to use these facts and derive other relationships. For example, since we are given the relationship of parents in the facts, we can derive a rule of grandparent. Similarly, we can count from it the number of children of a person.

dada(Dada, PotaPoti):-

baap(Dada, Abbu), baap(Abbu, PotaPoti).

countaulaad(Kitny, Waliden):-

bagof(Bachy, (aulaad(Bachy, Waliden)), Bachy),

count(Kitny, Bachy).

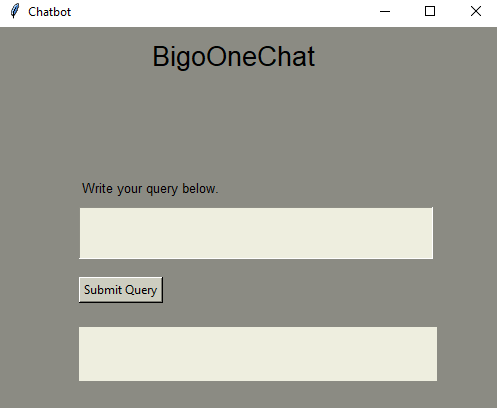
**User Query; Goal:**

The goal is simply a user query from the PROLOG file on the basis of facts and rules. Goal statements must be written on the same pattern in which facts are written and rules are defined. Prolog goal statement contains fact or rule name accepting parameters which can be constants or variables.



**How Chatbot works:**

Our chatbot is a simple python based graphical interface created in tkinter.



The chatbot expects user queries as Natural language. So, we have to convert it into a Prolog goal. For this, we used text preprocessing and useful text extraction.

**Text preprocessing:**

User query may contain some special characters which we do not want to process for example period, question mark. We discarded these characters from our query.

**Sample Query**: Who is the father of nadir?

**Output**: who is the father of nadir

**Useful Text Extraction:**

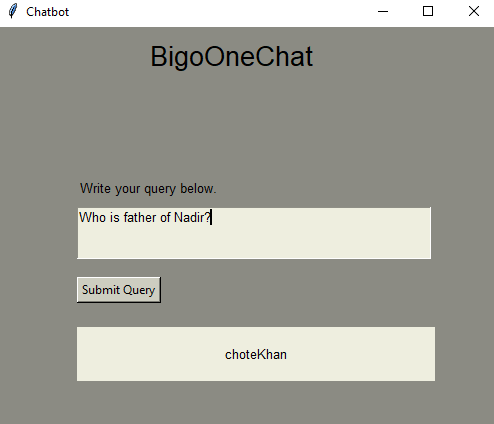
We extracted useful text from the query that is helpful in creating a Prolog goal. For example, rule names and constants etc. Proceeding with the above query, useful text is “father”, and “nadir”.

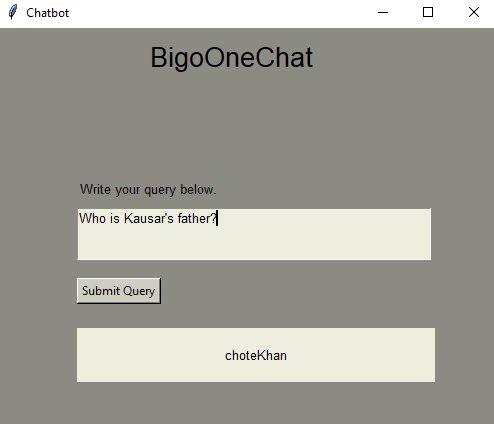
So, it generates the following **goal**: baap(Baap, nadir).

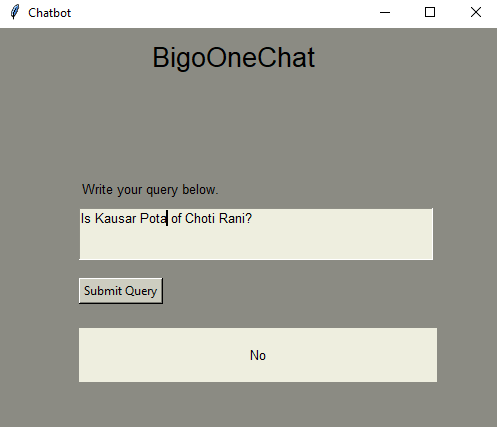
**Goal Statement**:

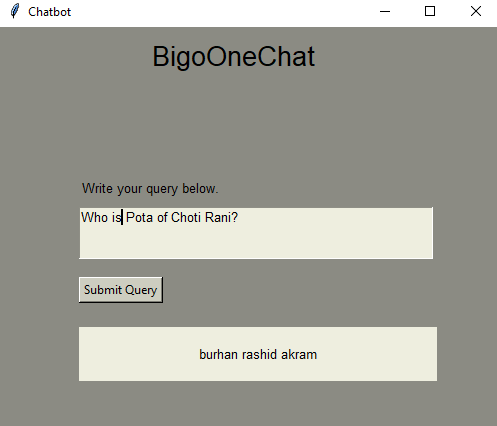
For our ease, we defined a template as “rule(X, Y)” which will be updated as the query processes in the text extraction section. Now, the goal statement is ready to be input to the prolog.query method. The chatbot will then process the query in accordance with the knowledge base file and return the result, which is rendered back to the Chatbot.

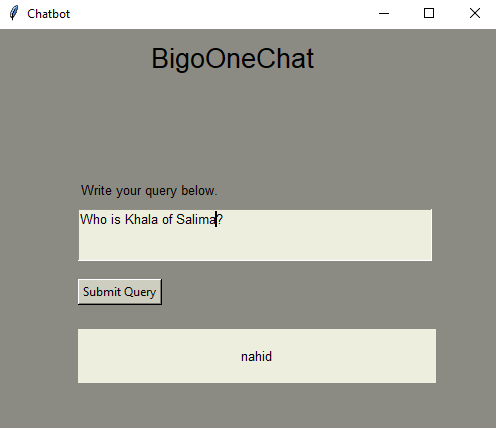
**Interactions with Chatbot:**

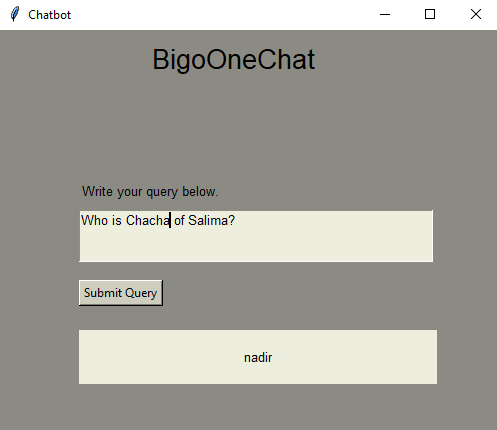
****

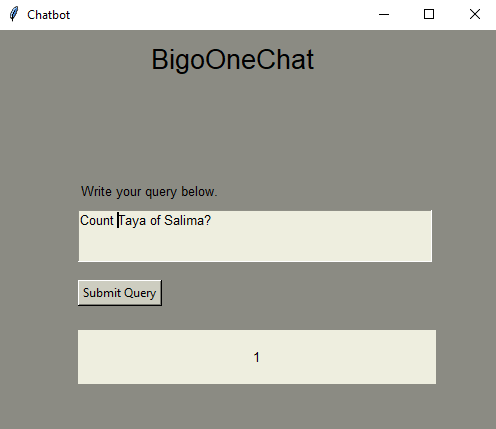
****

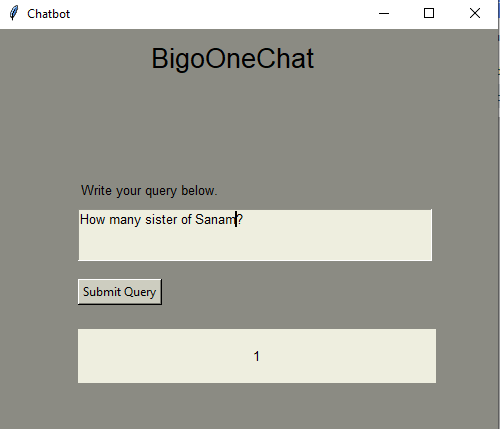
****

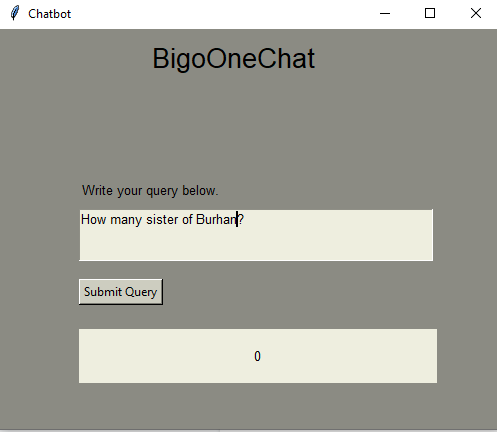
****

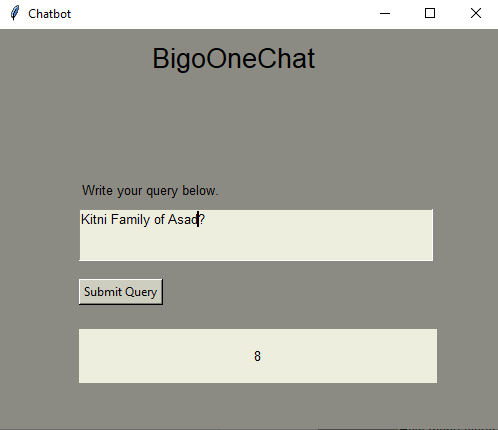
****

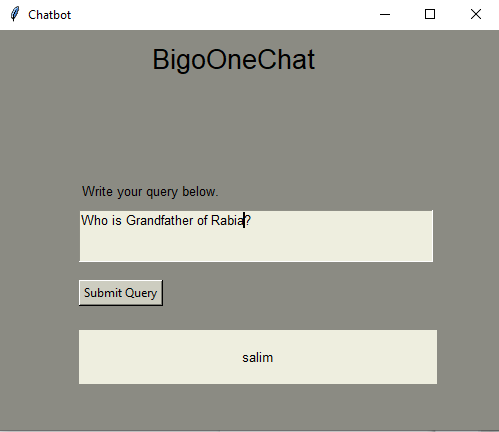
****

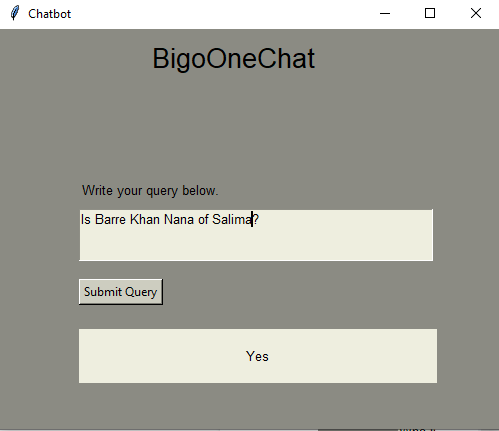
****

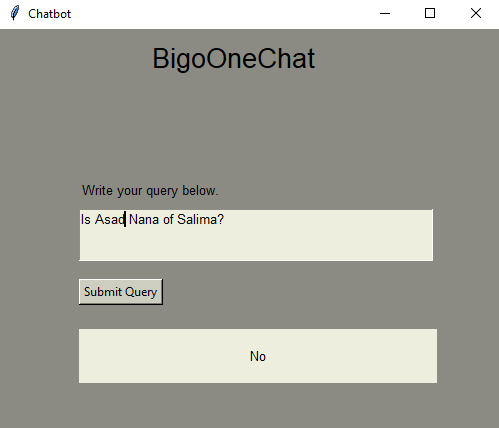
****

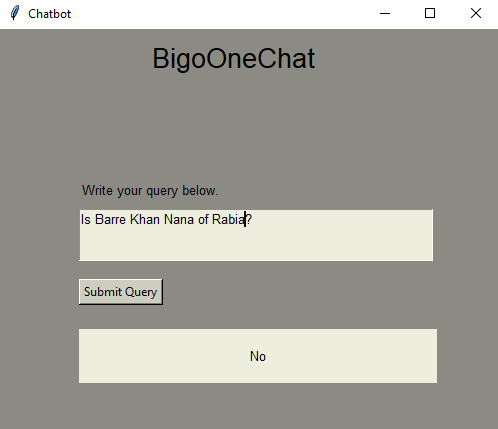
****

****

****

****

****

****

**Limitations:**

We have covered the following types of queries in our chatbot.

1. Who is “Relationship” of “Person B”?

(relationship(Relationship, Person B))

1. Is Person A has “Relationship” with “Person B”?

(relationship(Person A, Person B))

1. How many persons in this “Relationship” with Person B?

(countrelationship(Count, Person B))

There are other types of queries to be tackled like following;

1. What is the relationship of Person A with Peron B?

e.g. What is the relationship between Burhan and Akram?

Output: Brother

1. AND of two rules.

e.g. Is Burhan beta of Nadir and Nahid?