Peter Pham pxp180041

CS 4395.001

Karl the BirdBot

Introduction/Abstract

Using the NLP techniques I have learned thus far and the AIML library, I have created a very simple chatbot that the user can interact with and will remember their name, age, and their likes and dislikes. The bot will look for keywords in the user's input and will respond accordingly and will save the user data after the program has terminated.

Additionally, it has a database of knowledge that it can pull from so that it can give the user the correct information. It is all stored in a modified version of XML called AIML. The information was grabbed by a web crawler and is manually integrated with AIML so that the loading of the bot and the access of the information can happen as fast as possible.

System Information

The majority of the bot system is coded in AIML. AIML is a markup language that is derived from XML. The library downloaded from pip is either using an older version of python 3 or the library is built for python 2 but edits to the original library file are needed to make the program compatible with python v3.10 for some functionality not longer supported or changed. The aiml python library can create a kernel that can load the AIML files into its system and creates a bot with the input rules it was given. There are also options to save the compiled bot into what is called a "brain" file which can be recompiled at a faster rate than compiling with AIML files.

The program first tries to find the “brain” file to load in, if none exists, it then loads the pre-existing data files in the data folder. If there is a user data file, the program also loads it in and tells the bot about the user by setting its variable values. The program then introduces itself and its functionality, prompting the user for input. Once the user inputs something, the program proceeds to process the data.

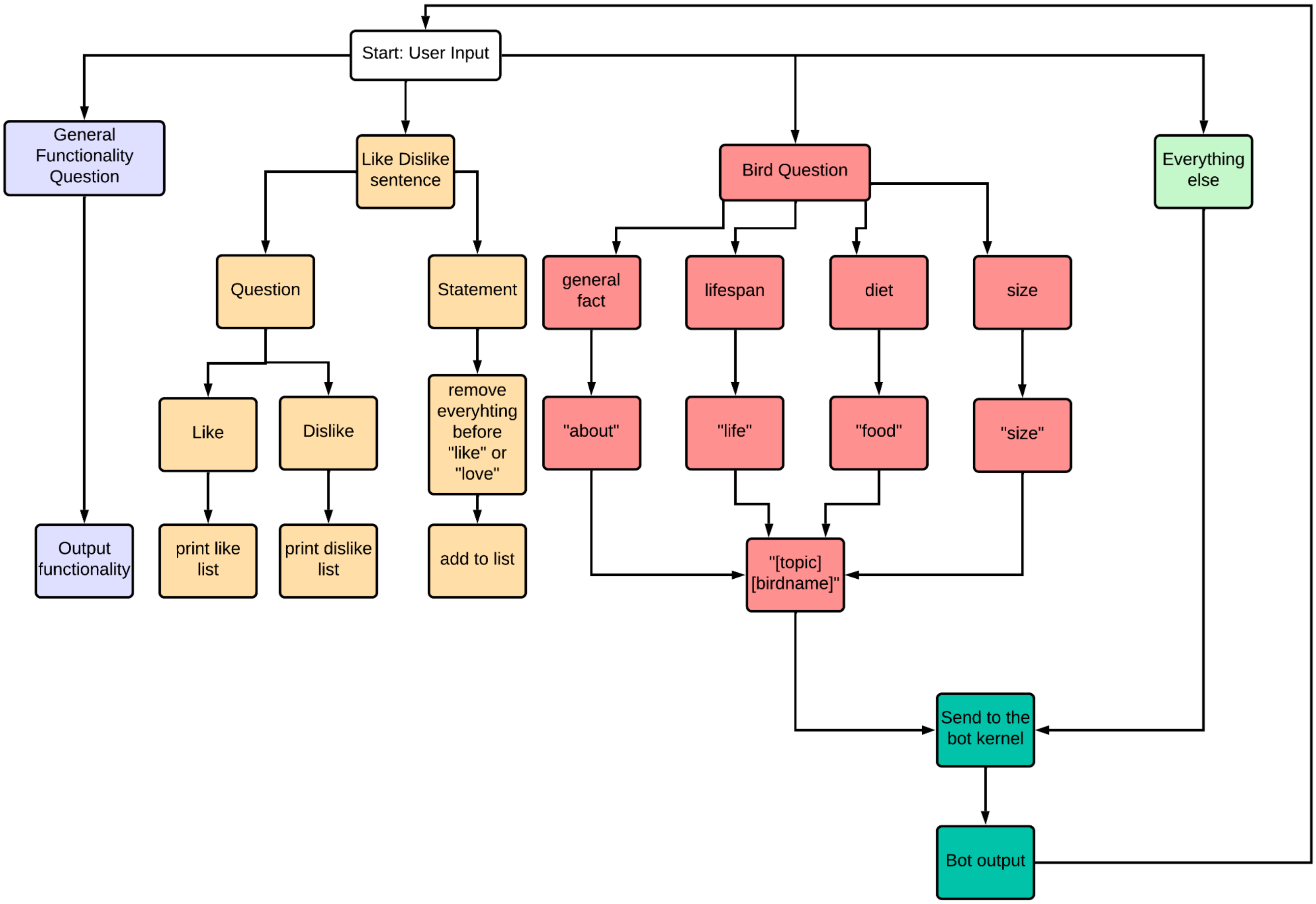
The processing of user data is coded in python utilizing the NLTK library to understand what the user wants the bot to do or say. Although there are functionalities built into the AIML language, it was too linear and yet complex for me to learn in a week and so I used base python to process some of the data so that I have an easier time coding the bot. The program first tokenizes the input and create a list of parts of speech tags with those token. From there, topics can be extracted from the input that is put into 3 different categories: a functionality question, a personal trait statement, a specific bird question, or a general statement.

Functionality questions are just general questions on the functionality of the bot. In this scenario, the program only uses the basic comparison method.

Personal trait statements are classified by the word that it contains. If the user enters any words relating to “like” or “dislike”, the system then uses sentiment analysis to decide whether the user likes or dislikes the item or action. If the user likes it, then the sentiment analysis comes back as positive, the opposite is true with dislike. In order to tell if the user is asking for their likes and dislikes versus telling the robot what they like, the “WP” part of speech is used to differentiate between a question and a statement

Specific sets of words are defined in order to tell the difference between what the user is asking. The main topics the program is looking for are the bird’s size, lifespan, and diet. A for loop iterates through the tokens and matches the token to the list of words. If there is a match a new response is created and breaks out of the for loop. Then the program looks for a bird name in the tokens. Once a bird name is found, the program concatenates that to the end of the new message and stores the bird name to a variable to be used later on if the user uses a pronoun. If no bird names are found but there is a part of speech tag with the label "PRP", the program uses the variable with the stored bird name instead and concatenates it to the new message instead.

If the input does not match any of the described topics above then the program just takes the raw input and feeds it to the bot. The general diagram of the processing logic goes like this:



If the users enter the word “quit”, the program then creates a custom-made person class. After the class is created, the program then grabs the user data from the bot and creates a Person class and inputs in all of the values that it knows. If the value is unknown, then it is set as an empty string or a None type which is saved as a pickle file to be read in at a later time. Then Karl says his goodbyes and the program closes itself.

Strengths

Karl is extremely good at giving bird facts. The web crawler has given Karl enough information so that random bird facts are unlikely to be repeated and specific bird questions are processed accurately. Karl is also very good at remembering user data and characteristics although there are not a lot of characteristics to remember.

Weaknesses

Karl is basically not very good at everything else. He does not make very good jokes, most edge cases are unable to be processed by Karl. If the user makes a spelling mistake, Karl will most likely not understand what they mean. If the user doesn't enter the entire name of the bird, Karl will not understand what the user wanted even though the bird's name may be unique. Karl’s knowledge of specific birds is only limited to 15 different bird species and Karl doesn't even know that much about those bird species.

Appendix

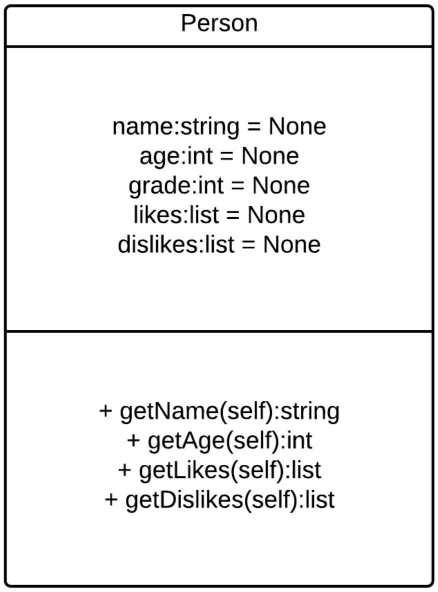
Appendix A: Code Source

All code sources for Karl the BirdBot are hosted on my personal GitHub page. Users can freely use it as they please.

Appendix B: Bird Data

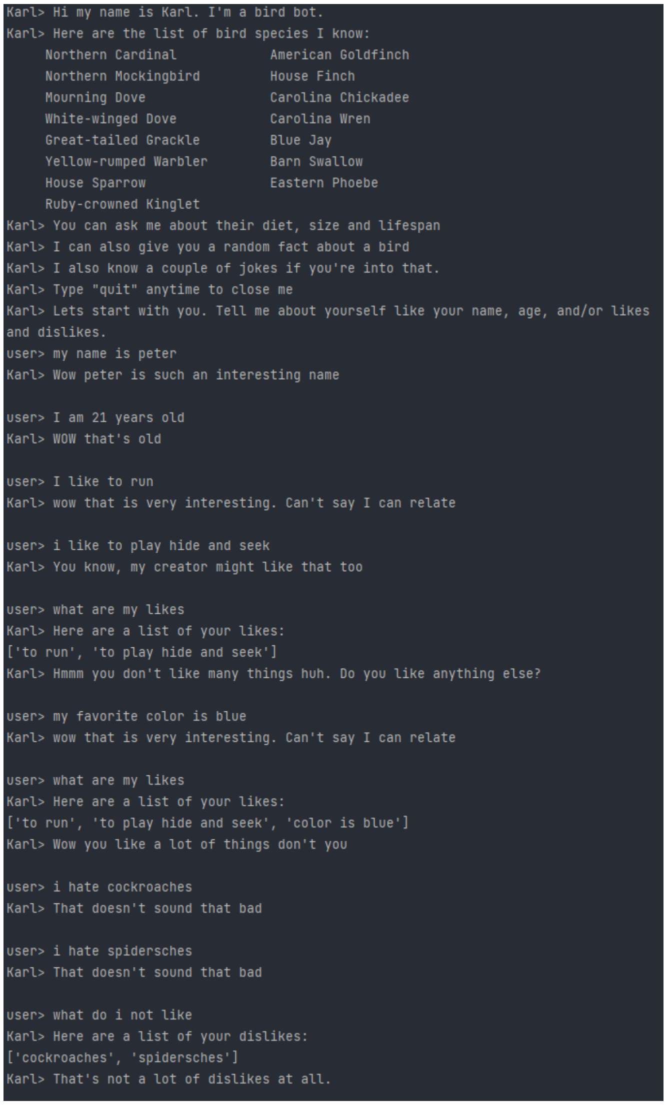
The list of data for the bird files is stored in the “birds.data” file included in the code file when downloaded. It includes a list of all bird facts Karl knows along with the things he knows about specific species.

Appendix C: User Data



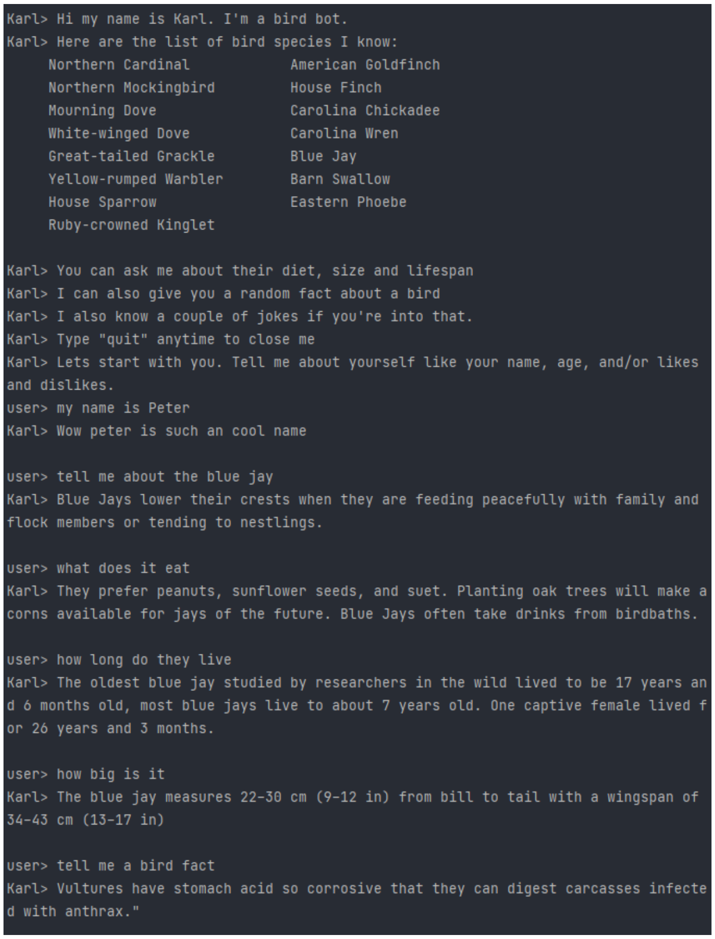
This is what the user data contains. Although the grade is a feature in the data class, I did not have enough time to implement it into the user chat processing step therefore it is left unused for now.

Appendix D: Sample User Interaction



```
Karl> Hi my name is Karl. I'm a bird bot.
Karl> Here are the list of bird species I know:
     Northern Cardinal            American Goldfinch
     Northern Mockingbird         House Finch
     Mourning Dove                Carolina Chickadee
     White-winged Dove            Carolina Wren
     Great-tailed Grackle         Blue Jay
     Yellow-rumped Warbler        Barn Swallow
     House Sparrow                Eastern Phoebe
     Ruby-crowned Kinglet
Karl> You can ask me about their diet, size and lifespan
Karl> I can also give you a random fact about a bird
Karl> I also know a couple of jokes if you're into that.
Karl> Type "quit" anytime to close me
Karl> Lets start with you. Tell me about yourself like your name, age, and/or likes
and dislikes.
user> my name is peter
Karl> Wow peter is such an interesting name

user> I am 21 years old
Karl> WOW that's old

user> I like to run
Karl> wow that is very interesting. Can't say I can relate

user> i like to play hide and seek
Karl> You know, my creator might like that too

user> what are my likes
Karl> Here are a list of your likes:
['to run', 'to play hide and seek']
Karl> Hmmm you don't like many things huh. Do you like anything else?

user> my favorite color is blue
Karl> wow that is very interesting. Can't say I can relate

user> what are my likes
Karl> Here are a list of your likes:
['to run', 'to play hide and seek', 'color is blue']
Karl> Wow you like a lot of things don't you

user> i hate cockroaches
Karl> That doesn't sound that bad

user> i hate spidersches
Karl> That doesn't sound that bad

user> what do i not like
Karl> Here are a list of your dislikes:
['cockroaches', 'spidersches']
Karl> That's not a lot of dislikes at all.
```

This is a sample of what it is like to introduce yourself to Karl. This section of user interactions is processed using sentiment analysis to determine if the user sees the noun or verb as positive or negative. Then the program assigns that as a like or dislike. It then outputs the lists as requested by the user.



```
Karl> Hi my name is Karl. I'm a bird bot.
Karl> Here are the list of bird species I know:
     Northern Cardinal            American Goldfinch
     Northern Mockingbird         House Finch
     Mourning Dove                Carolina Chickadee
     White-winged Dove            Carolina Wren
     Great-tailed Grackle         Blue Jay
     Yellow-rumped Warbler        Barn Swallow
     House Sparrow                Eastern Phoebe
     Ruby-crowned Kinglet

Karl> You can ask me about their diet, size and lifespan
Karl> I can also give you a random fact about a bird
Karl> I also know a couple of jokes if you're into that.
Karl> Type "quit" anytime to close me
Karl> Lets start with you. Tell me about yourself like your name, age, and/or likes
and dislikes.
user> my name is Peter
Karl> Wow peter is such an cool name

user> tell me about the blue jay
Karl> Blue Jays lower their crests when they are feeding peacefully with family and
flock members or tending to nestlings.

user> what does it eat
Karl> They prefer peanuts, sunflower seeds, and suet. Planting oak trees will make a
corns available for jays of the future. Blue Jays often take drinks from birdbaths.

user> how long do they live
Karl> The oldest blue jay studied by researchers in the wild lived to be 17 years an
d 6 months old, most blue jays live to about 7 years old. One captive female lived f
or 26 years and 3 months.

user> how big is it
Karl> The blue jay measures 22-30 cm (9-12 in) from bill to tail with a wingspan of
34-43 cm (13-17 in)

user> tell me a bird fact
Karl> Vultures have stomach acid so corrosive that they can digest carcasses infecte
d with anthrax."
```

This is a sample of what it is like to ask Karl facts about birds. All of the birds reside in Texas, therefore questions about location will come up as an error. Bird facts are randomized.

Peter Pham pxp180041

CS 4395.001

References

1. pemagrg1, “PEMAGRG1/easy-chatbot: Easiest way to create a chatbot using AIML. created date: 18 Nov 2018,” *Easy-Chatbot* , 18-Nov-2018. [Online]. Available: https://github.com/pemagrg1/Easy-Chatbot. [Accessed: 05-Apr-2022].
2. AIML-Foundation, “AIML docs,” *AIML Foundation*, 20-Jun-2018. [Online]. Available: http://www.aiml.foundation/doc.html. [Accessed: 09-Apr-2022].
3. P. Sharma, “Advancing AIML to build a chatbot,” *OpenGenus IQ: Computing Expertise & Legacy*, 19-Jun-2020. [Online]. Available: https://iq.opengenus.org/advancing-aiml/. [Accessed: 09-Apr-2022].