

Artificial Intelligence Neural Network

Software Version: 0.0.4

Paper by:

Dexter Rio Charles Shepherd

Problem

In the 21st century humans, lead busy lives. You don't have time to clean the house as you have to work and when you get home you have to cook. We get little time to ourselves.

If we are too busy and do not get enough rest research shows, we are more likely to have strokes and suffer breakdowns. Living in a messy house is also no way to live and can prevent you resting. Can we really be bothered to clean the house after a long day of work?

Elderly people need care, as do disabled people. When there is no family, around they need a carer but there are not enough carers around to cater to the persons need.

How can we do all these complicated jobs when we are too busy to do it, don't have enough people to do it and don't have the money for a servant?

The system in this paper is a learning algorithm, which learns specifically what the user teaches it. The system can apply its knowledge in similar situations. This system can help millions of people by making their lives easier.

The system can do whatever you can do as it learns through interaction. You can come home to a tidy home and a nice cooked meal every day so you can put your feet up and not have to worry about anything.

Another issue is that current AI on the internet is that people cannot teach it, cleverbot doesn't know who you are, how can you have a conversation with something which takes limited data in.

Planning and Design

Task

Building on V0.0.3, the AI, SHEP, in this version is edited to have many features to enhance its intelligence and ability to act like a human.

The user should have the ability to add layers of vocabulary to answer complex sentences, which may have multiple points and subjects. "bill and harry like cheese". Research will need to go into this to find an effective extra layering method. If nothing is found in the data for an extra layer, the code will use what it has to answer.

SHEP needs the ability to look at lots of data and solve problems. For example learn to play chess, and possibly get better. This could be done with the layer method, but then an extra function which allows the user to add modes to SHEP. "maths mode" would be added to SHEP then SHEP would learn keywords and what to do with them, through logic and teaching. "area of a square", keyword = "square", "ask side length", square answer.

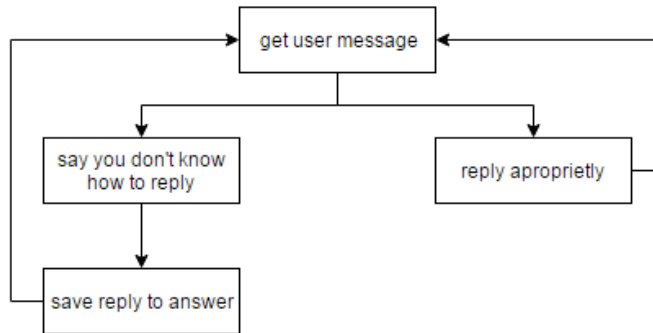
The above function may require a coding language of sorts within SHEP, to program SHEP's thoughts, it needs to be a simple code, with simple input, output, if statement and maths operations, loops shall not work. The coding language could be programmed through text input to make it easy to program.

Requirement

The AI will be able to take in data and output the most appropriate response it can think of. The AI is able to navigate through a network of files to find the output. The system can learn from the user's teachings, and then append this to its data.

SHEP is able to remember what is said in the sentence and possibly deceive a human into thinking he is human. To test that I will have to set up an input/output system using HTML code on a random chat website, but this is not for this version.

User Interface



Success Criteria

Test plan

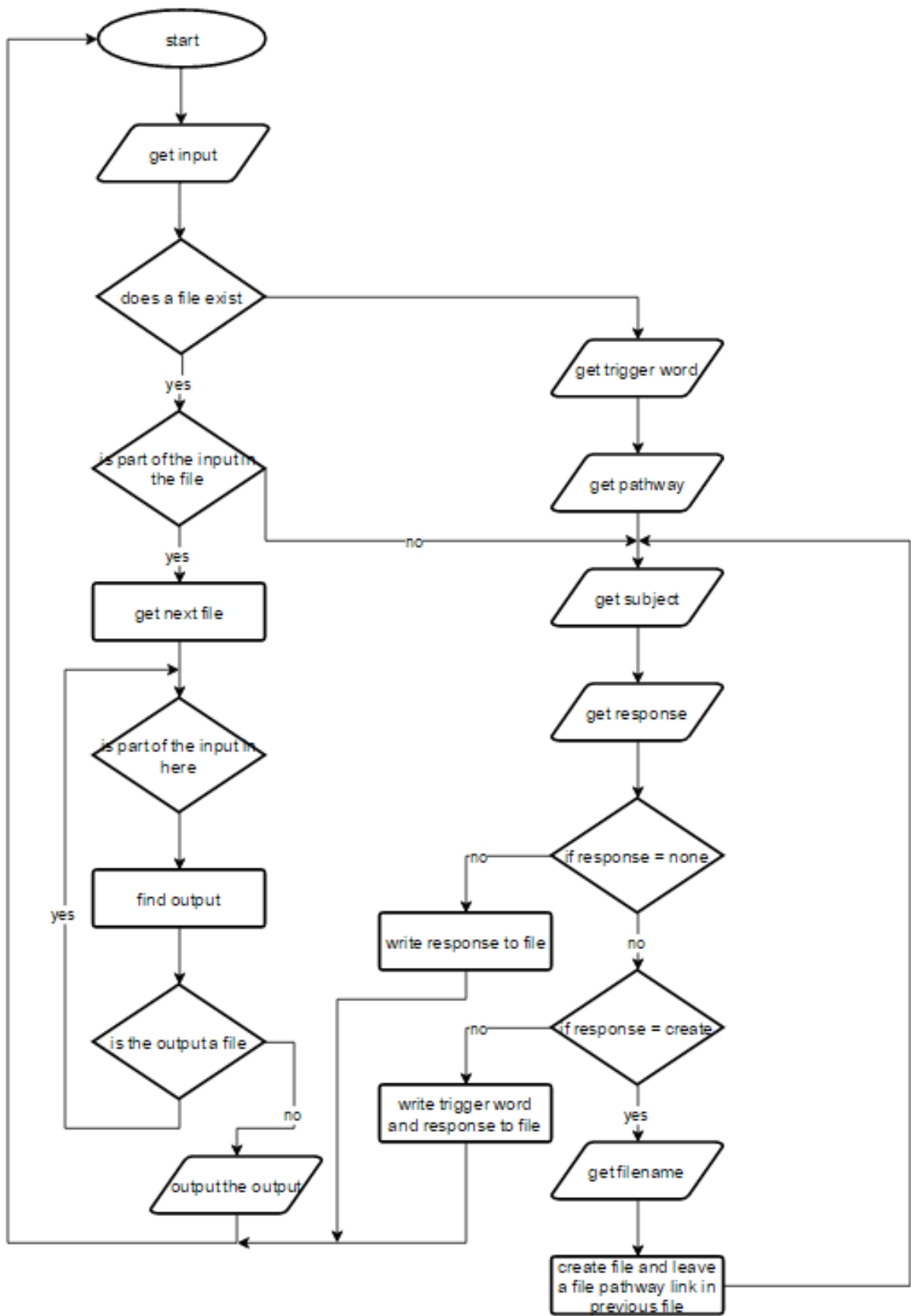
Test Number	Test	Expected result
1	The AI can have layers added to.	When the user adds a layer, it shows up in a layer list and is output at the beginning of every time. The user goes to edit > add layers and types in the layer name and it is added to a file.
2	The AI checks the first layer added and their words.	The layer opens up files and the sentence has its words checked to see if they are in the file.
3	The user is able to add words to a specific layer	The user goes to the menubar, and selects edit > add to layers. Then the user types in the layer name, and what word the want to add.
4	The AI will go down the layer route if there is a sentence containing the layer, and output the found sentence.	Words are added to the data, and it goes through an extra series to find an output.

5	The AI will not go down the layer route if vocabulary in the layer is not found.	It outputs whatever is in the data.
6	The AI is able to learn to solve problems, for example area of a square	You can use any parameters and get different outputs to the problem.
7	The AI' creates files with code to solve problems.	
8		
9		
10		

Variable Plan

Hardware

Flowchart



Pseudocode

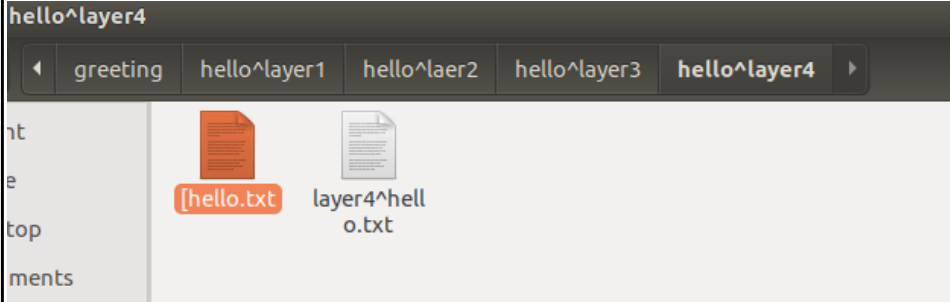
Coded Solution

Development

Development diary

Date	What I worked on
05/07/2017	<p>Today I worked on this paper, setting it up and thinking of ways to tackle the tasks. I made a folder for the version 0.0.4 containing the 0.0.3 code for edit, which has a few bits changed. All the "0.0.3" in the code was changed to "0.0.4".</p>
07/07/2017	<p>Today I worked on the code layer functions. You can now create layers, name the layers, and add vocabulary to them. This is done within two options, "add to layer" and "add layer".</p> <p>I came across many bugs, but these were user made bugs. For example; having "file" as the variable link instead of "f", after declaring "f = open....".</p> <p>I next need to add the ability to read the layers when searching for a response.</p>
10/07/2017 – 12/07/2017	<p>Within these days ia have been on holiday, so haven't been able to dedicate as much time. However, I have added functions to check for layerts, layer files ect... I have not completed it yet, but the code works as normal so far, while checking the layers.</p> <p>Next I need to finish the function so it can creates files for the layers and then finds the output. This will need to be tested, and planned.</p>
15/07/2017	<p>I have made it so the code loops round to find the layers and travel down directories to find the output. After days of bugs, I finally got the AI outputting the chosen response, hidden deep in its file system.</p> <p>I am yet to test it out with multiple layers, I added a loop in to keep cheking it, but I await the hours of debugging I am yet to have.</p> <p>The remains a bug, when I use a layered piece of data, along side</p>

	<p>another phrase to create the paragraph output I coded in within V 0.0.3, it comes to a run-time error. I will look into this tomorrow, as it is night now. I think I will have to some how reset certain variables within the loop. Again, I shall look into it tomorrow.</p>
<p>16/07/2017</p>	<p>Today I worked more on the looping function so multiple laers can be added. So far it creates the correct folders, but it comes to a run-time error when loading up the root of the file. I will continue with this problem, but I got distracted by the bug that normal (non layered) sentences would not work.</p> <p>After an hour of debugging, I managed to make the AI learn properly by rearranging where variables reset, and pathways saved. However, another issue arose when the layers reset everytime to not have any layers stored. This meant ou could ask SHEP one thing and then other inputs would cause a run-time error.</p> <p>Evertime I said a term with over 1 additional layer, it appeared with this problem:</p> <pre>Exception in Tkinter callback Traceback (most recent call last): File "/usr/lib/python3.5/tkinter/__init__.py", line 1562, in __call__ return self.func(*args) File "/home/dexter/Desktop/AI/V_0.0.4/code.py", line 1468, in cleanup4 add_data() File "/home/dexter/Desktop/AI/V_0.0.4/code.py", line 959, in add_data file = open(system_pathway +pathway +oldpath,'w') FileNotFoundError: [Errno 2] No such file or directory: '/home/dexter/Desktop/AI/V_0.0.4/assets//home/dexter/Desktop/AI/V_0.0.4/assets/greeting/hello^layer1/layer1^hello.txt[hello.txt]' >>></pre> <p>I am thinking, to solve this problem I will need to add in all the needed layer files. This could be done by appending the current directory.</p> <p>After lots of debugging, I finally got the the files to be sorted out. I am not at an error where apparently a file does not exist when to me it does. It turns out the folder of the trigger is not included for some reason. I shall have to include it so the file can be written and read to.</p>
<p>17/07/2017</p>	<p>I finally got the layers working... I think. The files all save correctly, and the code travels through the folder system and finds the output. But I fear it hasn't worked, most things end up not working.</p> <p>It turns out it doesnt work, I can no longer add normal data to SHEP, and the data will not add from no data at all (meaning files must already exist to be read).</p> <p>I re-did a lot of code and edited in lots of if statements, this gave back the ability to teach normal sentences. I have made it so it can add data again, it is the reading which is the harder part.</p> <p>I have found that I must make a file reading mode to check the next file. This can be done using the loops within the "learn" function, copied and pasted.</p> <p>After lots of debugging, I managed to get the AI learning through normal</p>

	<p>sentences, layer sentences and outputting the learned output. I will do further testing of this tomorrow, to make sure that it can handle any amount of layer, with any words.</p>
<p>21/07/2017</p>	<p>Today I worked on the update function, it was not updating layered items.</p> <p>After a lunchtime of bugs, I managed to reset variables so SHEP could update layered items and non-layered items.</p> <p>I then tried to teach a sentence with 3 extra layers. It came into an error. I am not sure why this has happened? The run-time error message makes no sense. I managed to fix this by resetting and adding to the directory variable each time the loop looped.</p> <p>I have fixed this entire problem, I now added 4 layers of "hello" and it all works. The only side effect is you must teach SHEP the how to output twice. Exmple:</p> <pre> user: hello shep: I do not know how to respond to that..... user:hi user:hello shep: I do not know how to respond to that..... user:hi user:hello shep:hi </pre>  <p>Above shows the files.</p> <p>After 1 week of development (7 days), I have decided to release the current version of the AI as a tester for anyone to use.</p>

Testing

Test results

Test number	Test	Pass/fail	Comment
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Future ideas

Evaluation