# Documentation For AI Chatbot

## Design notes

|  |  |
| --- | --- |
| General explanations of the system and its goals | The system can answer a user’s question about dogs and has data on specific breeds of dog. It also can discern a dog from any other animal being able to discern if the image is of a dog, this is done using image classification. This chatbot also has a text to speech feature that can be turned on or off which allows all outputs to be also played in an audio format using pyttsx3. As well as this we also have a special quiz function which uses the knowledge base to ask questions, it will tell you a breed and we need to discern whether it is a dog breed or not.  If the user fails to enter one of the specified commands it uses our cosine similarity functionality to check if we have any data on the topic in our Q&A.csv. If it finds a similar sentence it will reply appropriately.  Image classification is done using TensorFlow and Keras. These are used to create a machine learning model which can recognise dog images and say whether the animal in the image is a dog or not. |
| The system requirements | The user should be able to ask any question about dog breeds and receive an answer. The user should be able to use any predetermined commands to activate specific functions. If the command does not match one of the predetermined commands it will automatically reply if there is a similar sentence in the database. The user also can enter an image to use for image classification. |
| The employed AI techniques | We use a cosine similarity file which is accessed if the user has not entered a predetermined sentence and it will return the most appropriate one. This uses the nltk and aiml libraries. It converts the sentence into a value and matches the value to a question in a CSV containing various questions and answers. When it finds the most suitable one it will return the respective answer and output that to the user. If the question is not similar to any question in our CSV with a threshold of 30 percent it will return a response reading "Could not find any answer".  We use nltks resolution prover and expressions to create and read a knowledge base which has some logical links between specific values. E.g. dog is not cat. This is also then used in a quiz setting which asks questions about breeds based on the data we have in the knowledgebase. Any new data added to the knowledgebase will be added to this quiz.  We use tensorflow/keras for image classification. This is for recognising if the image is a dog. The model has been trained on 4000 training images of dogs and 1000 testing images to increase accuracy. We also trained against 4000 images of other animals to help discern when the image is a dog or not. This also had 1000 testing images. As recognising if the image is a dog is complex and they have many features especially between different breeds I opted to focus on Hyperparameter tuning. For the tuning I found that for the model I needed to use 100x100 images and use larger than normal kernel and pool sizes. When using a 64x64 and the average kernel and pool sizes the accuracy was 40-50% but when I had increased the detail in the images it allowed the accuracy to increase. Also, I realised that doing to many neurons would have a negative effect, so I opted for 128 dense layers. This achieved me an accuracy of 75% using 32 filters then 64 filters. I then used 64 filters for both and got an accuracy of 82% which was very good considering the complexity of this problem. In respect to overfitting, I also implemented an early stopping callback which stopped the model from training if there was no improvement in 5 epochs. This removed overfitting in a sense and allowed me to train this model as much as possible before overtraining the model. Since the model was wanting to return a true or false to it being a dog or not I used binary-crossentropy for this and a sigmoid activation on the final dense layer so it would check if it is a dog or if it is not.  The text to speech uses pyttsx3 which takes advantage of the built-in text to speech model in windows and uses this to convert the text to speech. Any output to the user is then also played as audio. This feature can be turned on and off. |

## Conversation log

|  |
| --- |
| Task A:    If a user inputs a question like this is will return a response using Wikipedia.    Above is a response using our cosine similarity. It recognises that we asked about labradors and returned some information on them.    Above is a result that produces a cosine similarity below 30% meaning it will return a response with a placeholder answer.    Above is how to activate/deactivate text to speech. It will play an audio saying “Text To Speech: On” when its entered and will play audio of any returned data to console until it is turned off.  Task B:    The user can check if certain things are true in this case we say that labrador is a dog and it confirms that based on the data it has in the Knowledgebase.    We can also tell it to remember that certain things have links. In this case we say that Shihtzu is another breed of dog and this is remembered in the Knowledgebase.    We can additionally tell it to link that something can not be something else, In this case we say that Persian is not a breed of dog.    It checks if something is incorrect as well and will say this. If it has no knowledge on the subject it will say so.    If a statement has a contradiction it will say so and not remember what it has been told.    Above is an example of this working using different formats like – “is not”, “is not a” and “is a”  We can use “a” as an optional input and it wont effect the format and will work the same.    Above is a quiz. To get this enter “quiz” and it will ask you a true or false question based on the knowledgebase. So, any data we have told it before can also be used in this quiz. It has a mix of cat and dog breeds and to get it correct you need to define whether the given breed is of a cat or dog.      Above is an example of a given breed being used in the quiz.  Task C:      When we enter what is this image it will prompt the user with a file dialog (NOTE: File dialog can appear behind other applications so be aware) and the user will select an image. Above we used an image of a dog and it is correct.      This is the same thing but using an image of a cat and it correctly guesses it is not a dog.      Above we use a dog in a costume, and it fails to detect it is a dog due to the costume. This is due to the 80% accuracy so we can expect this to happen.    If a non-image file format is used it will notify the user |