# Chatbot design planning document

-Elliot Harding (N0688119)

**Chosen topic area**

This project is the attempt at creating a dog breed information chatbot. The chatbot will attempt to provide its user with information on different dog breeds in a conversation like manner.

**List of requirements**

In order to achieve the aim of the project, a list of requirements must be met:

1. Perform conversation pleasantries.
2. Provide the user with information on requested dog breeds.
3. Provide the user on requested lists of groups of dog breeds (i.e. hounds, terriers, retrievers, etc…).
4. Provide user information on dog sizes.
5. Tell the user if a dog is a cross or pure breed.
6. Attempt to guess and provide information on the dog breed a user describes.
7. Tell the user dog related question-answer jokes.
8. Perform Wikipedia searches when it cannot provide requested information.
9. Receive a picture of a dog as input and guess its breed.
10. Provide a dog theme based ‘toy world’ for the user to interact with.
11. Provide a game to the user based on guessing the size of dog breeds, and to demonstrate an example playthrough of the game using a double deep q network.

**Description of individual modules**

In order to achieve the requirements above, the chatbot implementation will utilize various modules.

TF-IDF

This module utilizes the term frequency, inverse document frequency and cosine similarity equations in order to calculate the similarity of a sentence in comparison to a list of other sentences. This is to be used by the chatbot to compare users input with its data to understand the users input, and therefore give a better response. The implementation of requirements 2 to 7 will use this module.

AIML

The AIML module will be used by the chatbot to define the response rules to users input. For example, to return information about a specific dog breed when users request it. Each input run of the chatbot will be initially handled by the AIML. It will decide the next steps of the chatbot algorithm in order to produce the correct output, except however in the case of certain pleasantries and jokes in which their entire functionality will be handled by it.

Dog breed and information

Data pertaining to a breed’s name with its corresponding information will be stored within an CSV, like a 2D array. The chatbot functionality will use this data in order to achieve requirements 2 to 7. The DDQN model will also make use of this data for training and running of the game guesser (requirement 11).

Image classifier

The image classifier component of the chatbot will be used to achieve requirement 9 (receive a picture of a dog as input and guess its breed). This will be implemented using a convolutional neural network. The network will be generated and trained in a separate script, in which a dataset of dog images will be fed into the network. This network model will then be saved, and the chatbot script will load the model and utilize it for classifying the images input by its user.

Toy world reasoning system

This module targets requirement 10 (provide a dog theme based ‘toy world’ for the user to interact with) and will be implemented using the NLTK library. The bot will utilize the library to perform natural language processing and a dictionary to build up a concept of the toy world.

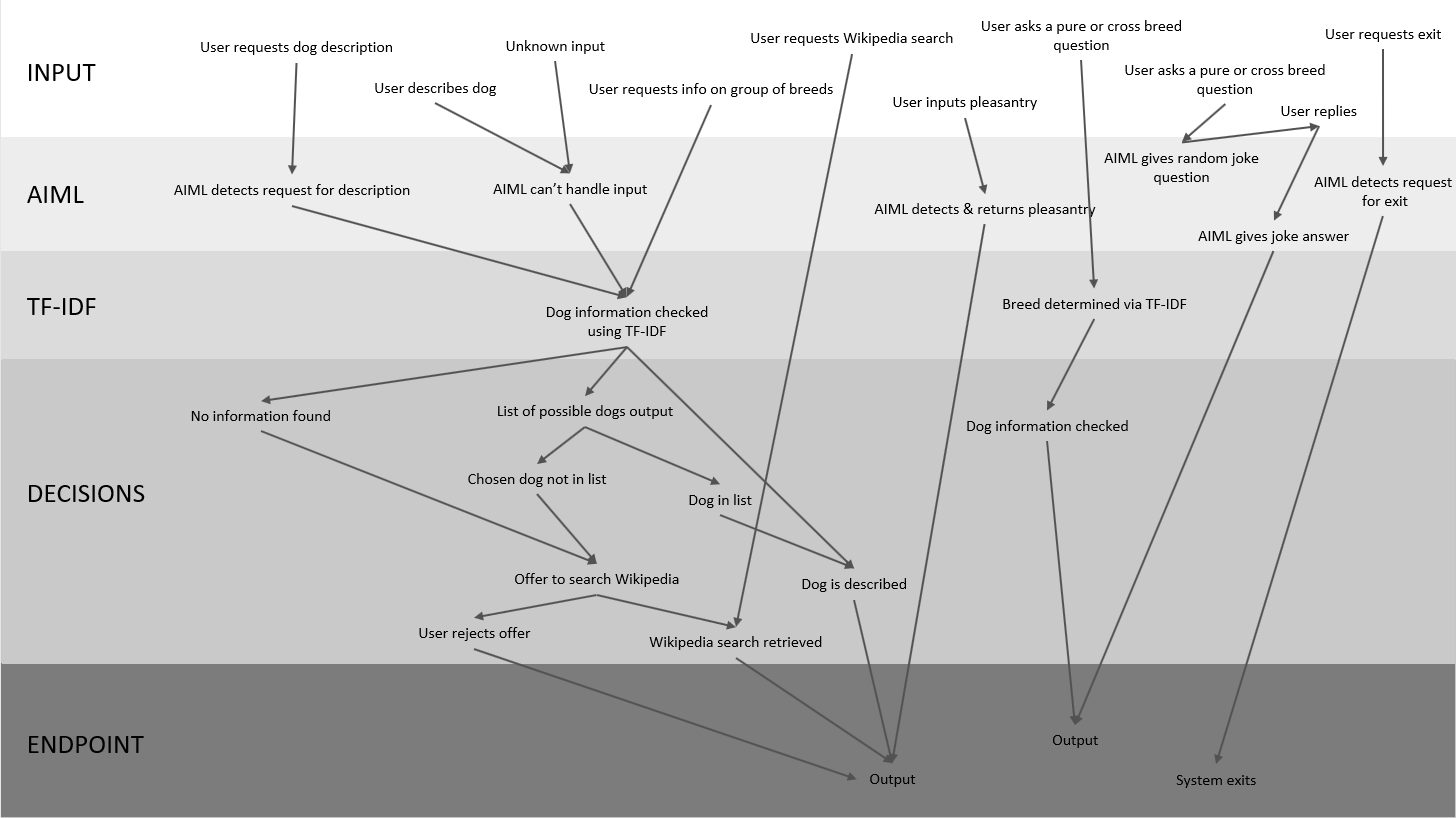
Transformer network

This module is used to deal with unmanaged user input by providing replies from a transformer network trained on a dataset of movie lines. The training will occur in another script in which the tokenizer (used for dataset formatting) and the weights of the transformer model will be saved for use by the chatbots own transformer model. Loading the pre-made tokenizer and save model weights will reduce the chatbots loading times significantly.

Double deep Q network game guesser

The double deep q network (DDQN) targets parts of requirement 11, to demonstrate a dog breed guesser game to the user. The network will be created in another script and trained on a dataset of dog sizes, its weights will then be saved for the chatbot to load into its own DDQN model which will be utilized to play the game.

**Flow diagram**

Below is a flow diagram representing the functionality flow between the various modules described above.

**Reflective document**

What went well, what not so well, work in context, what could have been done if there was more time. Training times, loading times.

PSEL issues

The imaging component of the chatbot takes pictures of dogs and attempts to predict the breeds of the dogs. A training accuracy of roughly 59% was achieved, this could have been improved if more training data/better resources/time was available.

Mention the fact you're loading the tokenizer from a saved version

mention that wiki thingy is still chill cuz wiki command