# Dataset generator & Chatbot assistant for Instruction Manuals

## Introduction:

Let it be an electronic whisk or a huge piece of machinery, instruction manuals play a major role in helping to understand the device. Although they are useful sometimes, they become a hassle when they are very long. The trainees or newly learning technicians spend a considerable amount of time on reading instruction manuals for various purposes. Our main aim is to reduce man hours invested on reading those instructions. It’s better to know the only required information instead of reading whole manual.

## Contents:

This project contains 2 codes - FinalCode.py & intent\_generator.py and a corpus.txt file.

* Corpus.txt:

This is an example manual. This is manual for a trimmer.

* Intent\_generator.py:

This python file takes corpus.txt as input and produces a JSON file.

* FinalCode.py:

This file creates chatbot and trains it with the JSON file provided by the intent\_generator.py.

## Process:

Firstly, the intent\_generator.py file is compiled using the corpus file. The corpus.txt file consists of data of the manual of a trimmer. This generates a JSON file [ in this case with name – file\_name.json ]. This consists of data as below

"intents":[

{"tag": "greeting",

"patterns": ["Hi", "How are you", "Is anyone there?", "Hello", "Good day"],

"responses": ["Hello, thanks for visiting", "Good to see you again", "Hi there, how can I help?"]

},

{"tag": "goodbye",

"patterns": ["Bye", "See you later", "Goodbye"],

"responses": ["See you later, thanks for visiting", "Have a nice day", "Bye! Come back again soon."]

},

{"tag": "thanks",

"patterns": ["Thanks", "Thank you", "That's helpful"],

"responses": ["Happy to help!", "Any time!", "My pleasure"]

}

]

Now the JSON file is taken as input for the FinalCode.py file. Here the chatbot is trained with the data present in the JSON file. As per the data when a user asks question it compares it with the tag, patterns and provides the responses from it as answers.

## Working:

For this project the language we chose is python.We downloaded some modules. For working of the code – Intent\_generator we used the modules NLTK for tokenizing, json for creating jsonfile, LexRank for text summarizing. But in case of the FinalCode.py we used numpy, tensorflow, tflearn, random, pickle, and also nltk. While downloading the tensorflow we faced some issues regarding GPU. The execution part is showed using jupyter notebook for this code for easy understaning.

1. Run the intent\_generator file from command prompt.

A screenshot of a computer

Description automatically generated with medium confidence

A Json file with name given in the command prompt will be generated and saved at the same path where the code got executed.

2. Now run the FinalCode.py. We have done using jupyter notebook.

Graphical user interface, text

Description automatically generated

Above is the screenshot of training data.

## Graphical user interface, text, application Description automatically generated

When user Interacts with chat bot replies as above.

Below are some conversations b/w user and chat bot.

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application

Description automatically generated

There is an error over here.

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Even though results are good, it has given wrong output for the input like – “how to Environment”.

## Further changes:

* We can make little bit changes like training the module more accurately.
* We can develop a GUI for interaction purposes.

## References:

ChatBot : <https://github.com/simranjeet97/ChatBot_Tensorflow_NLP>

## Team members:

Dinesh Kumar T

Adeshvar K Mullath

Abhinaya Sri Ginne