A Research Paper on HUMAN MACHINE CONVERSATION USING CHATBOT

ROHAN TIWARI[1], G.M.S.S. PRANAV[2], Mrs. R. PREMA[3]

[1]B.E Graduate (IV year), Department of Computer Science and Engineering, SCSVMV, Kanchipuram
[2]B.E Graduate (IV year), Department of Computer Science and Engineering, SCSVMV, Kanchipuram
[3]Assistant Professor, Department of Computer Science and Engineering, SCSVMV, Kanchipuram

---***---

Abstract – Chatbots square measure canny specialists with those clients will hold discussions, regularly by means of text or voice. Lately, chatbots became inescapable in organizations fixated on customer administration. In spite of a rising interest for chatbots in instruction, clear information while heading to style them as astute mentors has been scant. This paper presents a conventional philosophy for planning and carrying out a chatbot as a canny coach for a college level course. The approach is based upon first-request rationale predicates which can be utilized in different industrially accessible devices and spotlights on two stages: information reflection and displaying, and discussion flow. As the principle aftereffect of this exploration, we propose numerical definitions to display discussion components, thinking processes and conflict goal to formalize the philosophy and make it system free.

Key Words: Chatbots, Knowledge Modelling, Methodology, Conversation Design, Intelligent Tutoring.

1. INTRODUCTION

Tech-giant Google describes the word " Chabot" as: " A computer program designed to simulate conversation with human users, especially over the internet. " While communication remains the primary purpose of a chatbot, during this project, we realized that there is much more happening beyond the communication interfaces. The interfaces may change, but the underlying technology remains the same and that is AI. Therefore, though we have titled the project as" Human- Machine conversation using a Chatbot", this is essentially a project in AI. AI, Natural Language Processing and Machine learning are the core concepts involved in this project. Artificial intelligence is the ability of machines /computers to do things and make decisions based on their oven knowledge, without explicit instructions. This is a rather generalized definition of AI; it includes planning, understanding language, coming up with appropriate responses and learning from previous responses/ data, recognizing objects and sounds and problem solving. Machine Learning (ML) is a way in which AI can be implemented. It concentrates on training the system and letting it thins when it's over. The results, in this case, are based on the data using which the system is trained and the training itself. Deep Learning (DL) is an approach of implementing ML. It is inspired from the working of the human brain, how neurons are interconnected in the Kran to form a network. Deep Learning involves the use of Artificial Neural Networks, which mimic the working of the new ions in our brain. Natural Language Processing (NLP) is a very important part of AI. Without NLP, AI would not have been possible. NLP allows developers to organize and give structure to Knowledge.

1.1 SCOPE OF THE PROJECT

This project uses technologies of artificial intelligence and machine learning to create a chatbot. The chatbot is trained using data sets of various types so that it can answer a broad range of questions. It is different from a normal human machine interaction because the chatbot learns as it progresses and adapts quickly. The programming language that has been used in this project is Python. Though Java is still widely used for many real-time applications, Python remains one of the versatile and robust languages in modern computing. The fields of machine learning and AI are amongst the current trends in computer Science and Technology. Every year, new technological breakthroughs are announced in these fields. Research is actively going on in these domains. However, these are also not completely developed and are still "learning". Everyday there is something new our chatbot can learn. Therefore, there is no fixed standard for chatbots

2. LITERATURE SURVEY

●In the literature, the authors used different approaches to get different techniques and methods to get a better chatbot.

●For these reasons the existing systems had proposed many applications.

●Chatbot is an AI software. It can communicate with a user in natural language through messaging applications, websites, mobile apps. Conversation between human and automated conversational agents are more challenging. In this paper, I reviewed some literature that uses different techniques for chatbot implementation.

●M. D. Leonhardt et al. presented an approach, a chatbot is a program that simulates typed conversation, with the point of, fooling a human into thinking the human is talking to someone else. A chatbot is a communication software that simulates typed or voice communication with the human, including the purpose of fooling into thinking a human to talking to another person. A network management-aware chat-bot is created named as Dorothy, it manages the network by management protocols so since SNMP

●Recently a considerable amount of promising work has been conducted in chatbot design. O. V. Deryugina [2] presented a detailed survey on the history of the chatbot, their applications, and the first designs of such systems. We have noticed that the chatbot related research is mainly distributed in the following areas, (I) different approaches (e.g., retrieval and generative), (ii) length of the conversation, and (iii) according to the domain.

●B. A. Shawar and E. Atwell [2] described a system to access Arabic language information using chatbot without sophisticated natural language processing or logical inference. They showed this work properly with English and European languages and how the same chatbot will react in terms of Arabic

3. PROJECT DESCRIPTION

Proposed method:

●The proposed system is designed to overcome the problems faced in day-to-day life by people. In our absence also we can react to the other people who are contacting us.

●The proposed chatbot is designed to take advantage of the portability of mobile devices and provide a simple user interface that makes use of the system easy for each user.

●Compared to other systems, our systems provide a mobile and compact design.

Knowledge modelling:

This phase determines how knowledge is represented and stored in the knowledge base. Conversation flow. Both the lexicon used by the tutor and the order in which ideas are presented should be defined in this phase. This section presents the formal definitions and foundations of the proposed methodology first. Later, each phase is described and contrasted with real-life queries. Finally, since implementation details vary across different conversation frameworks, they are not covered per se in the methodology. Nevertheless, conflict resolution and good implementation practices are broadly presented along with a case study.

We use following to make a chatbot using python.

Intents. Json – The data file which has predefined patterns and responses.

train_chatbot.py – In this Python file, we wrote a script to build the model and train our chatbot.

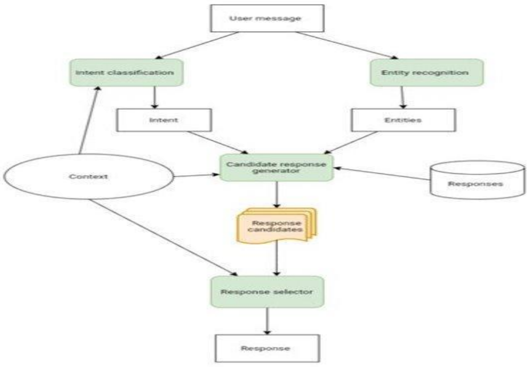
Words. PKL – This is a pickle file in which we store the words Python object that contains a list of our vocabulary.

Classes. PKL – The classes pickle file contains the list of categories.

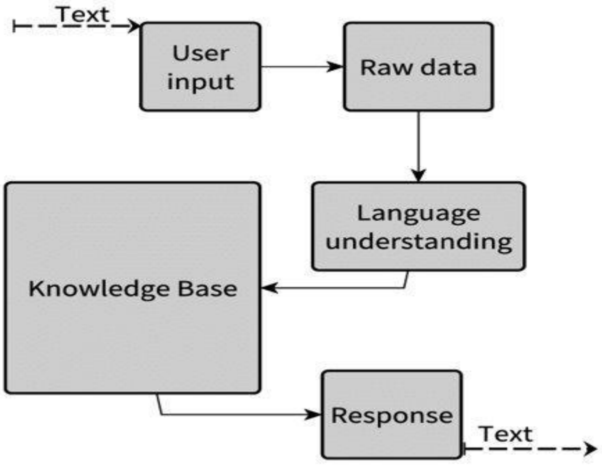
Chatbot_model.h5 – This is the trained model that contains information about the model and has weights of the neurons.

Chatgui.py – This is the Python script in which we implemented GUI for our chatbot. Users can easily interact with the bot.

Process:



Architecture:



Algorithm:

•Prepare the chatterbot library using chatterbot corpus command.

•Now we have an environment and a chatterbot library to work on and now we have the following to do:

The pilot program: The pilot program is the main or core code of the program through which the chatbot runs on the system. In addition to this we have the bot trainer and the bot predictor which will be discussed later in detail.

• Firstly, in the pilot program the system and TensorFlow are imported.

• Then we import Bot Predictor and then give the instructions to the bot to run and then when the questions are asked to the chatbot let it learn how to detect the question and find appropriate answers for them to print for the user.

•The Bot Trainer and The Bot Predictor are also called the support modules as they help the pilot program to work and execute properly.

PROJECT DESCRIPTION:

We have developed 2 models of chat bots. One is a social media chat bot or assistant deployed on Facebook. The other is a simple command line-based chat bot that can be used for small talk which also has speech output. The essence of this project remains to provide a simple demonstration of the building of a basic chat bot (while getting hands on practice on ML and AI concept which presents in self as an exciting and interesting field to us). The world is increasingly moving towards AI and chatbots and they are the future of human machine interaction. Chat bots are soon expected to take over traditional user interfaces. The chatbot on Facebook can be personalized as per the users or companies' requirements. It can act as a tool to increase interaction among the user and Facebook friends. The command line-based chat bot, which we have named "Spinzer'', doesn't require Internet connectivity to interact effectively with the user. It can also tell stories, jokes, capitals of countries, perform basic arithmetic and indulge in human-like conversations. It has been trained on multiple datasets. The dataset that this chat bot was trained on was made after cleaning and censoring the conversations to make the chat bot polite and formal. The following pages of the document will display the system architecture and other diagrams related to the project along with algorithms. We use Python, TensorFlow, PyCharm according to the present information we have, and it can be changed in the future accordingly.

System architecture:

The system architecture tells us about how the data is divided into categories and how the response generator generates response and then how the response is selected from the responses which we have given to the chatbot and then finally the response is given to the user.

Use case diagram:

This tells us about how the bot user, the Facebook user, collects the data and basic instructions from the data sets and reacts accordingly.

Class Diagram: The class diagram gives us a basic knowledge of how the classes are divided in the code and how they run with the instructions given in them and interact.

The Bot Trainer:

The bot trainer trains the chatbot with the datasets uploaded and keeps training the bot as we keep adding datasets to it. The chat bot learns new things as we keep using it and this plays a major role in the response selection.

•Firstly, we need to import tokenized data and then import a model creator.

•Then we use some instructions or commands to give the input to the chatbot and train itself for generating responses that is self-training using Sequence-to-Sequence.

•The bot runs on the loop for training itself.

•Still the information is being gathered on this and will be updated when known properly.

•The chatbot is trained in such a way that when it detects the same question again it tries to improve the level of response and make it more precise.

•The main task of the bot trainer is to make the chatbot more precise and flexible to the data.

•We use the logic adapters command to list adapters used to train the chatbot.

For example: "Chatterbot. Logic. Best Match"

•Then we create some strings for communication purposes and use "List Trainer" and train the chatbot with them.

The Bot Predictor:

As the name suggests it predicts what response should be given to the user. This part of the code uses the tokenized data and from that collects the relevant information to process. Then the loop is run throughout the dataset, and it sets a score to every response it gets when the loop is complete the response with highest score is given as the output to the user from the chatbot.

•Firstly, we import NLTK, string, TensorFlow and then we import tokenized data, knowledge base, and some libraries for checking pattern and remaining call functions.

• Then we create appropriate classes and objects to carry out the whole process of prediction and response generation.

Examples: def predict_ class (x, y)

def get response (s, d, f), etc.

After all these are carried out then we give the basic instructions to the chatbot that what should be done when the user is satisfied and greet the user and exit the execution of code.

Program to tokenize the data:

This process is used for turning a meaningful piece of data into a random string of character caller tokens. They can be used as reference to the original data but can't be used to get the original values.

It can also be defined as the program to turn sensitive data to non-sensitive data called tokens that can be used in a database or internal system without referring to it.

In simple words splitting larger forms of text into smaller forms or lines or words.

•Firstly, for this program we need to import TensorFlow, named tuple and some other libraries to begin with.

•For example:

Consider the string: "This is a cat."

After tokenization we get: ['This', 'is', 'a', 'cat'].

We can use this tokenized form to count the frequency of the word and number of times a particular word is present in the data set.

•Using PyCharm to execute code for chatbot:

•After installing the latest version of PyCharm we can add the code to it and use manage.py and run our code in the required area. This can only be done after you have done the settings part and filled all other requirements. At the end we need to have a code for only debugging purposes and that should be separately kept until required.

Components Used:

Hardware Requirements

1.System: Nvidia GPU/AMD

2.Hardisk: 00gb

3.RAM :4gb

4.Microphone Speakers

Software Requirements

1.Operating system: windows/Linux/mac OS

2.Coding language: python

3. Libraries: Tensor flow

4.Ide: Anaconda (PyCharm)

Working model and Implementation:

•First install all the required modules and libraries.

•This project consists of a chatbot made to function on PC's, without internet.

•This can be ideal for organizations and companies which want to stay connected to their customers all the time.

•It can also increase communication and can be seen as an interesting aspect to talk with an AI when used for personal accounts.

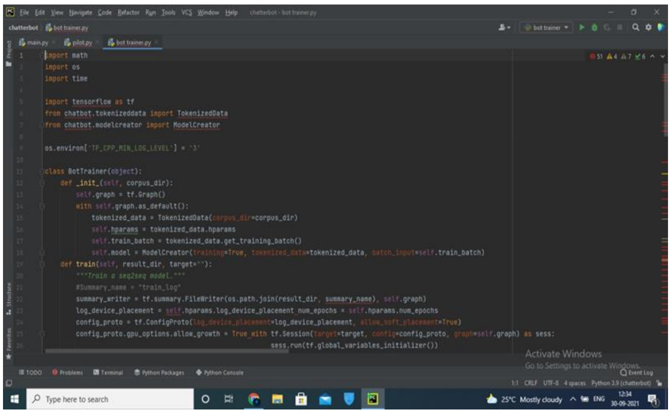
•The console based chatbot was the one which took the most effort, because it was trained on a cleaned data set and cleaning and Censoring words, etc. required a lot of effort.

•A lot of effort would also be required to train the model which took us several hours.

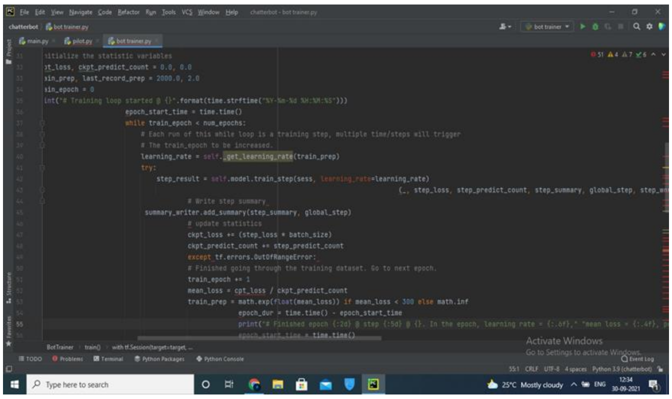
•The feature that stands out here is the ability of the chatbot to reply in human voice. This has been added to facilitate communication in a better way.

4. RESULT

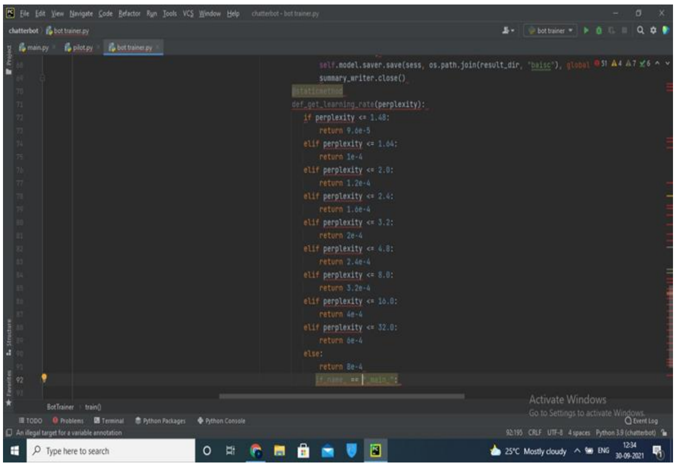
For Bot trainer:


File Edit View Navigate Code Refactor Run Tools VCS Window Help chatterbot - bot trainer.py
chatterbot bot trainer.py
bot trainer
main.py pilot.py bot trainer.py
TODO Problems Terminal Python Packages Python Console
Event Log
1:1 CRLF UTF-8 4 spaces Python 3.9 (chatterbot)
Type here to search
25°C Mostly cloudy ENG 12:34 30-09-2021
Activate Windows
Go to Settings to activate Windows.

```
import math
import os
import time

import tensorflow as tf
from chatbot.tokenizeddata import TokenizedData
from chatbot.modelcreator import ModelCreator

os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'

class BotTrainer(object):
    def _init_(self, corpus_dir):
        self.graph = tf.Graph()
        with self.graph.as_default():
            tokenized_data = TokenizedData(corpus_dir=corpus_dir)
            self.hparams = tokenized_data.hparams
            self.train_batch = tokenized_data.get_training_batch()
            self.model = ModelCreator(training=True, tokenized_data=tokenized_data, batch_input=self.train_batch)
    def train(self, result_dir, target=""):
        """Train a seq2seq model."""
        #Summary_name = "train_log"
        summary_writer = tf.summary.FileWriter(os.path.join(result_dir, summary_name), self.graph)
        log_device_placement = self.hparams.log_device_placement num_epochs = self.hparams.num_epochs
        config_proto = tf.ConfigProto(log_device_placement=log_device_placement, allow_soft_placement=True)
        config_proto.gpu_options.allow_growth = True with tf.Session(target=target, config=config_proto, graph=self.graph) as sess:
                                                  sess.run(tf.global_variables_initializer())
```

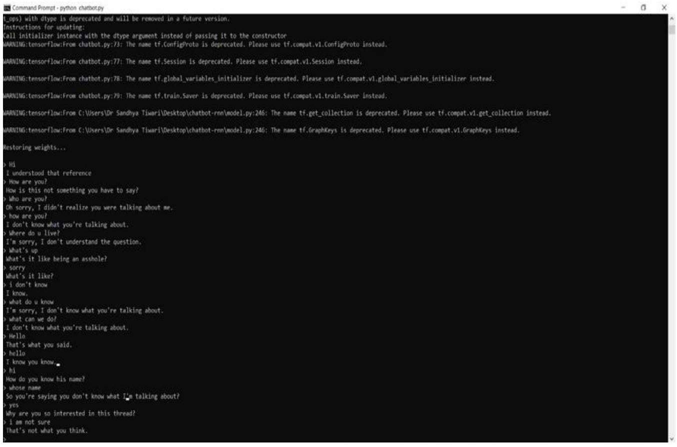


```
itialize the statistic variables
pt_loss, ckpt_predict_count = 0.0, 0.0
ain_prep, last_record_prep = 2000.0, 2.0
ain_epoch = 0
int("# Training loop started @ {}".format(time.strftime("%Y-%m-%d %H:%M:%S")))
                    epoch_start_time = time.time()
                    while train_epoch < num_epochs:
                        # Each run of this while loop is a training step, multiple time/steps will trigger
                        # The train_epoch to be increased.
                        learning_rate = self._get_learning_rate(train_prep)
                        try:
                            step_result = self.model.train_step(sess, learning_rate=learning_rate)
                                                                              (_, step_loss, step_predict_count, step_summary, global_step, step_wo
                                  # Write step summary
                         summary_writer.add_summary(step_summary, global_step)
                                  # update statistics
                                  ckpt_loss += (step_loss * batch_size)
                                  ckpt_predict_count += step_predict_count
                                  except tf.errors.OutOfRangeError:
                                  # Finished going through the training dataset. Go to next epoch.
                                  train_epoch += 1
                                  mean_loss = cpt_loss / ckpt_predict_count
                                  train_prep = math.exp(float(mean_loss)) if mean_loss < 300 else math.inf
                                          epoch_dur = time.time() - epoch_start_time
                                          print("# Finished epoch {:2d} @ step {:5d} @ {}. In the epoch, learning rate = {:.6f}," "mean loss = {:.4f}, p
                                          epoch_start_time = time.time()
```



```
            self.model.saver.save(sess, os.path.join(result_dir, "baisc"), global
            summary_writer.close()
    @staticmethod
    def_get_learning_rate(perplexity):
        if perplexity <= 1.48:
            return 9.6e-5
        elif perplexity <= 1.64:
            return 1e-4
        elif perplexity <= 2.0:
            return 1.2e-4
        elif perplexity <= 2.4:
            return 1.6e-4
        elif perplexity <= 3.2:
            return 2e-4
        elif perplexity <= 4.8:
            return 2.4e-4
        elif perplexity <= 8.0:
            return 3.2e-4
        elif perplexity <= 16.0:
            return 4e-4
        elif perplexity <= 32.0:
            return 6e-4
        else:
            return 8e-4
            if __name__ == "__main__":
```

This is the result we get after training the bot and also adding bot prediction code to it. The given below picture is an example of how our chatbot works and replies to us.



```
t_ops) with dtype is deprecated and will be removed in a future version.
Instructions for updating:
Call initializer instance with the dtype argument instead of passing it to the constructor
WARNING:tensorflow:From chatbot.py:73: The name tf.ConfigProto is deprecated. Please use tf.compat.v1.ConfigProto instead.

WARNING:tensorflow:From chatbot.py:77: The name tf.Session is deprecated. Please use tf.compat.v1.Session instead.

WARNING:tensorflow:From chatbot.py:78: The name tf.global_variables_initializer is deprecated. Please use tf.compat.v1.global_variables_initializer instead.

WARNING:tensorflow:From chatbot.py:79: The name tf.train.Saver is deprecated. Please use tf.compat.v1.train.Saver instead.

WARNING:tensorflow:From C:\Users\Dr Sandhya Tiwari\Desktop\chatbot-rnn\model.py:246: The name tf.get_collection is deprecated. Please use tf.compat.v1.get_collection instead.

WARNING:tensorflow:From C:\Users\Dr Sandhya Tiwari\Desktop\chatbot-rnn\model.py:246: The name tf.GraphKeys is deprecated. Please use tf.compat.v1.GraphKeys instead.

Restoring weights...

> Hi
 I understood that reference
> How are you?
 How is this not something you have to say?
> Who are you?
 Oh sorry, I didn't realize you were talking about me.
> how are you?
 I don't know what you're talking about.
> Where do u live?
 I'm sorry, I don't understand the question.
> What's up
 What's it like being an asshole?
> sorry
 What's it like?
> i don't know
 I know.
> what do u know
 I'm sorry, I don't know what you're talking about.
> what can we do?
 I don't know what you're talking about.
> Hello
 That's what you said.
> hello
 I know you know.
> hi
 How do you know his name?
> whose name
 So you're saying you don't know what I'm talking about?
> yes
 Why are you so interested in this thread?
> i am not sure
 That's not what you think.
>
```

4. CONCLUSIONS

This project was an attempt to understand and implement the power of AI, ML. We have learnt a lot during this project and aspire to take up research in the field of artificial intelligence.

Like mentioned before, chatbots are the future. We are making huge progress in this field every day. All the biggest computer tech companies around the world are looking up to AI. The great scientist Stephen Hawking had predicted that the world's first trillionaire will come from AI (that is, by making AI products).

The chatbots in this project can be used on any system by making the python programs into executable files particular to that system. It can also be changed into a particular or specific chatbot as the user requirements in the future.

REFERENCES

1. Al`es, Z., Duplessis, G.D., Șerban, O., Pauchet, A.: A methodology to design human-like embodied conversational agents. In: International Workshop on Human-Agent Interaction Design and Models (HAIDM 12). Valencia, Spain (2012), https://hal.archives-ouvertes.fr/hal-00927488

2. Anderson, J.R., Boyle, C.F., Reiser, B.J.: Intelligent tutoring systems. Science228(4698),456–462(1985).

https://doi.org/10.1126/science.228.4698.456,

http://science.sciencemag.org/content/228/4698/456

3. Angeli, A.D., Johnson, G.I., Coventry, L.: The unfriendly user: Exploring social reactions to chatterbots. In: Proceedings of International Conference on Affective Human Factor Design. pp. 467–474. Asean Academic Press (2001).

4. Dutta, D.: Developing an intelligent chat-bot tool to assist high school students forlearning general knowledge subjects. Tech. rep., Georgia Institute of Technology (2017).

5. Goel, A.K., Polepeddi, L.: Jill Watson: A Virtual Teaching Assistant for Online Education. Tech. rep., Georgia Institute of Technology (2016).

6. M. J. Pereira, L. Coheur, P. Fialho, and R. Ribeiro, "Chatbots' greeting to human-computer communication,"arXiv preprint arXiv:1609.06479,2016.

7. O. Deryugina, "Chatterbots," Scientific and Technical Information Processing, vol. 37, no. 2, pp. 143–147, 2010.

8. J. S. Malik, P. Goyal, and A. K. Sharma, "A comprehensive approach towards data preprocessing techniques & association rules," in Proceedings of The4th National Conference, 2010.

9. B. A. Shawar and E. Atwell, "Chatbots: are they really useful?" in LDVForum, vol. 22, no. 1, 2007, pp. 29–49.

10. A. Bordes, S. Chopra, and J. Weston, "Question answering with subgraph embeddings," arXiv preprint arXiv:1406.3676.

BIOGRAPHIES

"ROHAN TIWARI, pursuing fourth year in bachelor's degree in Computer Science and engineering department in SCSVMV University, Kanchipuram

"G.M.S.S. PRANAV, pursuing fourth year in bachelor's degree in Computer Science and engineering department in SCSVMV University, Kanchipuram"

"Mrs. R. PREMA, Assistant Professor in the Department of Computer Science and Engineering, SCSVMV University, Kanchipuram."