International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 01 | Jan 2021 www.irjet.net p-ISSN: 2395-0072

INTELLIGENT COLLEGE ENQUIRY CHATBOT USING TENSORFLOW

Ahmed Abdullah[1]

[1]*Ahmed Abdullah, Student, Department of Computer Science and Engineering, Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad-500090, Telangana, India*

Abstract - *This project is aimed to develop a python based intelligent chatbot using Natural Language Processing libraries in Python so that the chatbot can interact with the user. It is very difficult for the students who live very far away from the college or other non-college people who have to come to the college to ask basic enquiries. People have to travel to long distances to enquire about information that is not available on the college website. This chatbot will allow them to enquire about various details like upcoming events, faculty information, etc., eliminating the need to go to the college specifically to enquire or calling up the staff to ask them. Along with it, the user will also have the option to converse with the chatbot on any topic if they desire. The chatbot are able to answer any specific college enquiry queries as well as have a conversation with the user about almost anything by identifying user's intent.*

Key Words: **Chatbot, Deep NLP, Enquiry, Conversation, Intent**

1. INTRODUCTION

The fields of data science and artificial intelligence have progressed rapidly over the past decade. The ability to think creatively like a human is what makes the systems developed with AI more beneficial from other systems. Over 73 percent of the businesses around the world prefer to use AI to assist them if it can help them to gain profit and sustain themselves. More than three quarters of the consumers worldwide use chatbots as technical support for their customers as these chatbots are able to efficiently solve the customers' problems eliminating the need for human workers thereby cutting the expenditure of a large number of companies.

A College enquiry bot is very essential these days for providing information to non-college/college students about the college. Many students do not have any information regarding the college and the information that is present online is very less and hence, less people know about the working of college and students become very uncertain whether to take admissions or not. To overcome this problem, an intelligent chatbot is created which answers the users queries in any form of input. The chatbot understands the user's message, what type of information is being enquired and then it answers the query accordingly. The user can ask queries and the chatbot analyses the questions, interprets the meaning using a tag and intent and provides an answer.

There are two types of chatbots that are classified using the purpose of their design and the type of information they will provide to the user.

- Retrieval Based Chatbots – These chatbots fetch the input question and its answer directly from the database where the question's context is already predefined. These chatbots are very easy to build using simple NLP libraries of Python.
- Generative Based Chatbots – These chatbots are very hard to build as they require hundreds of conversations between users based on which they train themselves to predict their own answers.

1.1 GENERATIVE BASED CHATBOT:

This chatbot is based on a generative based chatbot. The generative chatbot is not built using predefined responses and instead, is based on a number of human conversations about college related information. They are open domain, meaning they are not confined to any goal. They are also able to make small talk with the user. They require large conversational data to train themselves. A generative chatbot focuses to perform machine translation methods and translate from an input to an output response based on an LSTM learning approach to predict the answers. The conversations which are very lengthy are more difficult to process and create answers.

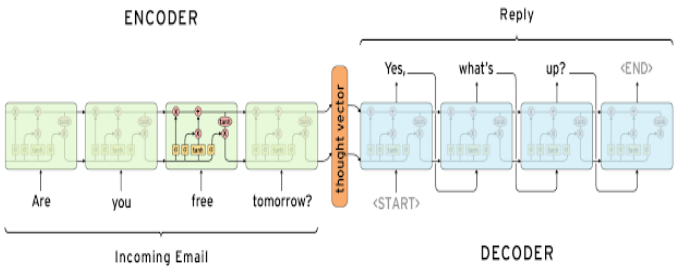

Fig-1: Example of Encoder-Decoder Working [1]

Using TensorFlow API 1.5, An executable Python file is created with the required code to execute the chatbot. The chatbot is trained separately with a dataset of college-based questions and answers which enables it to construct its own sentences. Generally, purpose specific chatbots are only useful if the input query by the user is exactly the same as in the database. However, if the chatbot is constructed using NLP, it can understand any form of sentence construction and generate a correct answer.

1.2 APPLICATION

The chatbot will be trained by a number of queries of different users. It will contain different information about various activities being held in the college and their dates as well as information about the college, faculty and the campus itself rendering it useful for both students and the non-college people. The chatbot can be shared or put up online on any of the college platforms such as the official college website where everyone is given access to download the chatbot. Anyone who has python installed on their personal computers or laptops can execute the chatbot directly which is pre trained and converse with it asking all their queries. As new and new information will be needed to be provided to the people over the course of time, that information can be added to the database, get the chatbot trained again and its updated version can be posted on the website.

2. PROPOSED METHOD

Firstly, a dataset has to be created containing the required questions and answers about the college information. The chatbot requires at least 500,000 conversations for it to understand the semantics and grammar. Any general conversation dataset like Cornell's Movie Corpus, Reddit corpus, Twitter corpus and so on can be used for this. The data set has to be modified and a minimum of 1000-1500 conversations have to be added in that data set regarding various college related enquiries.

The chatbot is then trained on this dataset for generally 100 epochs with the aim of reducing the loss error rate and increasing the accuracy. In an ideal situation, the accuracy would be 1.000. and loss error would be 0.000.

For execution, When the user runs the chatbot, the chatbot dialog box opens and user can enter its query and submit it. Once the query is submitted, the query is sent for data preprocessing. The data is cleaned of various symbols, numbers and stop words are removed. Every word is mapped to its root word and put in the same category for easy revival. Higher form of English such as the words, I'd, I'll and so on are converted to their base form. This cleaned data is converted into a bag of words model where every word is given a unique integer id to be used later during the answer construction process.

The checkpoint model i.e., the trained data is loaded into the chatbot and a prediction/thought vector is created. This context/thought vector that generates the data based upon the previous conversations. The prediction vector predicts the values with the help of the neural network and the output is generated word by word which is displayed to the user in the chatbot dialog box.

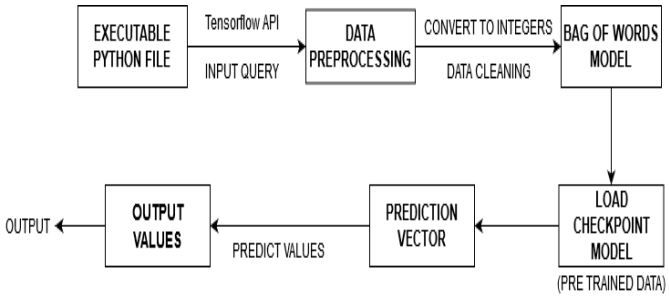

Fig-2: Outline of the proposed method

Fig.2 represents the working of the proposed college enquiry chatbot application. The chatbot creates its own sentence so the user can speak in any way and get the required information from the chatbot.

The chatbot consists of 3 layers which make it able to easily analyse and interpret the answer very quickly within minimal time. the output is predicted using the test prediction function with which a learning rate and a keep probability rate is associated. The words present in the integer form having higher weights are given more priority and selected for prediction. The output is more accurate when the training loss error is less and accuracy is more. The prediction is made on the most likely word that comes after the previous word and if it is correct, the chatbot predicts the next word and checks the loss if it is being reduced. If not, then it will move on to the next word and make the prediction again. The inputs can be of any form and any structure. However, if a question pertaining to a specific topic is asked, the chatbot will provide incorrect answers as the generative chatbot supports only general everyday conversations and not discrete conversations. For instance, if questions like," What is the width of a human brain?", "How many people live in Kukatpally area?", "Are there glaciers on Pluto?" cannot be answered because they are discrete topics and not something people converse about every day. Such specific enquiries should be inserted into chatbot dataset and then the chatbot should be trained with the dataset so it can answer specific questions easily.

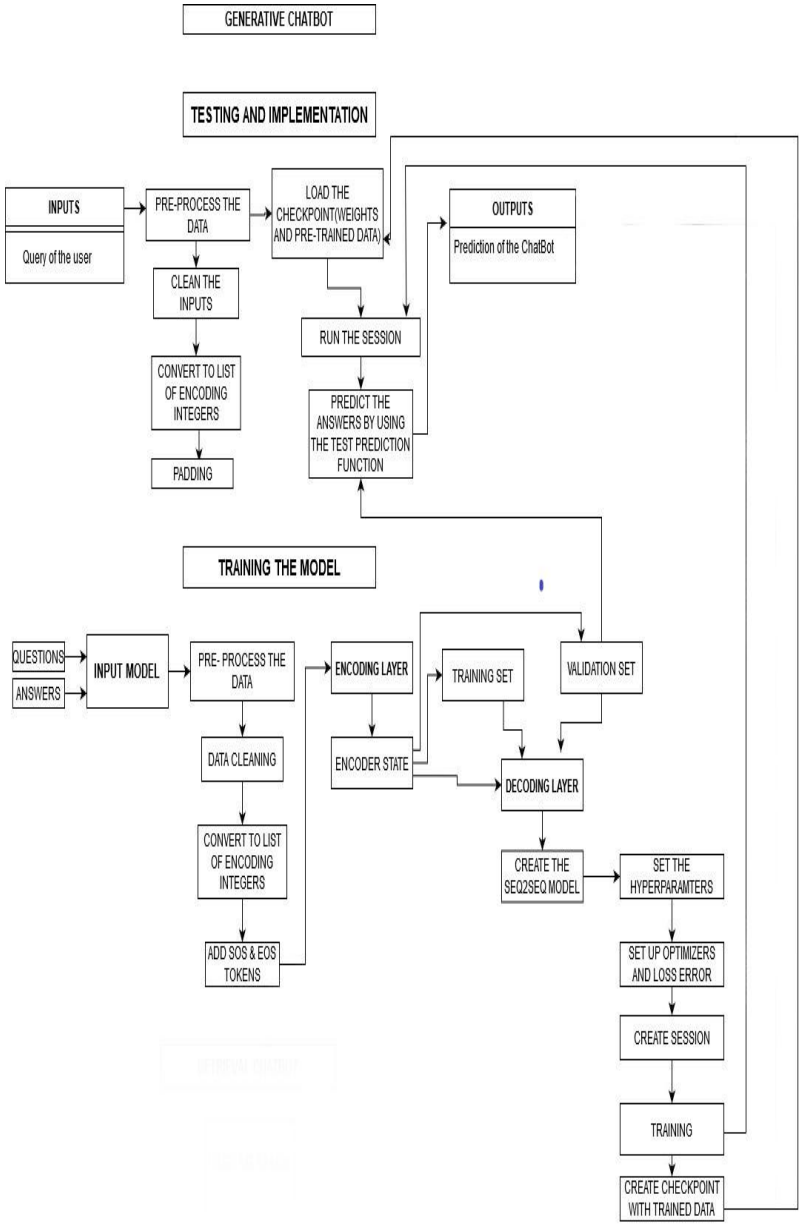

Fig-3: Block Diagram depicting the Process Flow

Fig.3 represents the step by step working of the chatbot which contains the training and the testing phases. The chatbot data is split and used for training and testing phases usually in the ratio of 60:40, but it can be changed by setting the keep probability rate.

3. ARCHITECTURE

There are several layers that make up the architecture of the bot model. They are:

1.) INPUT LAYER
2.) ENCODE LAYER
3.) ATTENTION MECHANISM
4.) DECODE LAYER
5.) OUTPUT LAYER

→ REQUIRED NUMBER OF HIDDEN LAYERS

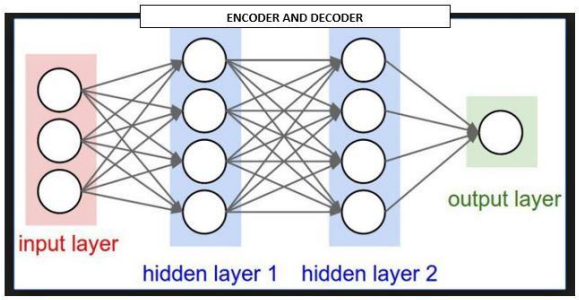

Fig-4: Neural Network Layers [2]

3.1 INPUT LAYER

The input layer consists of input data which is inputted by the user. It consists of artificial neurons which are responsible for data transfer to the next layer. This is the first layer of the artificial neural network. It transfers the input data in the form of integers to the next hidden layer where the data is multiplied by the initially set low weights. It receives the data backpropagated by the output layer back to this layer where it again propagates it forward with updated weights.

In the input layer, the input nodes provide the data to the hidden layer from the input given by the user. The multi-layer perceptron will be used to construct the chatbot. This will consist of few hidden layers that will allow the chatbot to understand the data better and improve the accuracy of the model.

3.2 ENCODE LAYER

The encoding and decoding of the architecture are a part of the RNN through which the data is translated, processed and given a meaning which can be later used to create vectors to predict answers based on user queries. This is the usual way through which the sequence-to-sequence predictions are generally made. This model used the seq2seq learning method using the mechanism of attention. Long Short-Term Memory networks is used to read the input data and understand the data. The data is then converted into vector form called prediction vector exactly like a bag of words model where every word is converted into an encoding integer. There can be any number of hidden layers in an encoder-decoder model based on the complexity of the model. The more complex the model, the more hidden layers are present and more time is required to process the data sequences.

The method of predicting an answer, understanding what comes next is called as language modelling. The new output is dependent on the output gained before. A set of LSTM cells are created where every LSTM cell accepts a single input. The input is a word in the encoded integer format. After the input is taken, then the information about that word is taken after which that LSTM cell is propagated forward.

Every word here in the list created is of the order a_b, where a is the word and b is its order. The hidden state of the word which is h_b is then calculated using the formula:

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$$

This will allow the encoder vector to get the hidden state, also called as the encoder state. The encoder state is the output that we get from the encoder that contains the information about the input processed in the encoder in accordance with a particular learning rate, dropout rate and the current weights applied. This encoder state then becomes the input of the next part of the neural network i.e., the decoding layer. The decoding layer receives the output of encoding layer (A vector called context/thought vector) and takes it as its first input along with few other parameters such as decoder cell matrix, batch size and RNN size.

3.3 ATTENTION MECHANISM

The attention mechanism is a very essential component of the neural network. The attention mechanism was first created for NMT (Neural Machine Translation) by the use of sequential-to-sequential networks. These models consist of encoder and decoder layers which form the neural network and create the seq2seq model which predicts the data. There is a huge problem when the context vector is generated at the end of the processing of the encoding layer. The context vector is not designed to remember lengthy sequences. The context vector forgets the already processed input sequence when it processes very long sentences. So, the attention mechanism was introduced to combat this difficulty.

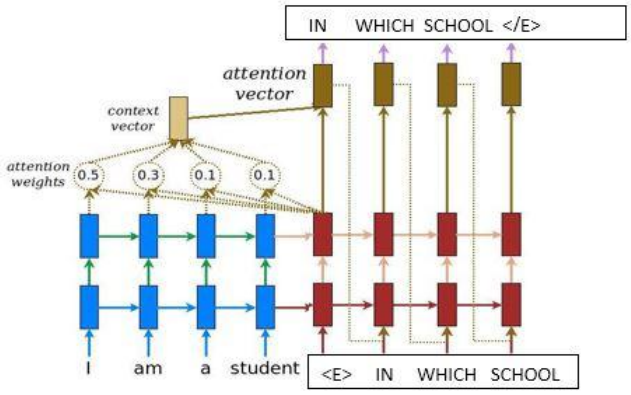

Fig-5: Working of Attention Mechanism [3]

Attention vector is a vector that uses the SoftMax function which assists in determining the final output. The encoder looks at the entire input sentence and the creates the thought vector. But when the sentence is long, the attention vector makes the encoder ignore the rest of the lengthy sentence and instead, focus on the important keywords of the sentence and then generate an output for the current keyword it is processing. When the keyword has been processed, it proceeds to the next word and generates an output sequence. This attention mechanism is only used when processing longer sentences which can create difficulty in learning for the context vector.

There are certain steps by which the attention mechanism works:

1. The encoder is fed the information containing the lengthy sequence.
2. The words are converted into vectors and stored in a list.
3. The list is propagated through the neural network.
4. SoftMax function is applied to the vectors along with the attention weights.
5. A context vector is then generated and the output of the SoftMax function is added to it.
6. The output generated by the decoder is the next correct word in the sentence.
7. Same is repeated for the following words of the input sequence.

The attention function, optimization and score function is different for every neural network and it is important to create a correct attention mechanism for the neural network to effectively predict the answers.

3.4 DECODER

The decoding layer is a part of the RNN network that processes the data received in the form of vectors from the encoding layer and produces the output. The output of the encoding layer i.e., the encoder state is used as a main component in the decoding layer where using the prediction vector, the most likely possible prediction is made according to the rate of accuracy. The output is a fixed vector that shows how the input sentence is processed. The loss is obtained which along with the output which is then passed onto the final output layer.

The decoding layer gets the encoding state as its input which decodes a probability of what word is best to be predicted using the SoftMax function.

SoftMax function gives back a probability function of each output word that can be chosen. The result that is achieved in the SoftMax function is rounded off a value between 0 and 1. Whichever value is the greatest, it will be chosen and that output is predicted.

Along with the SoftMax function, the bias, the weights and the hidden encoder state are all taken as inputs and then finally the prediction is made based on the previous outputs. The final output is generated and then sent to the output layer for displaying on the screen/GUI.

3.5 OUTPUT LAYER

The output layer displays the output along with the loss generated. This loss generated is a gradient of the loss function generated every epoch. This error/loss is backpropagated back through the hidden layers onto the input layer. The initial weights are updated in accordance with the loss and the whole sequence is repeated again with the new updated weights. This is called as an epoch. These epochs are done until the loss reduces to a very low amount and the accuracy increases to an optimum of 95 percent.

4. MODULES

4.1 Importing and pre-processing the dataset

The data is cleaned by separating the questions and answers are separated and mapped to each other.
4 special tokens are created that perform various tasks on the dataset:

1. PAD – PAD token is created to ensure that the length of questions and the answers remain the same.
   Eg: Question: What is your name? – length – 4
   Answer: It is Abdullah <PAD> - length – 4
2. SOS – The SOS token indicates the start of a sentence so that the neural network knows a new sentence has begun in the dataset.
3. EOS – The EOS token indicates the end of a sentence signalling the Neural Network to look for the next sentence by finding the next SOS token.
4. OUT – If a word is not present among the vocabulary in the dataset, the OUT token is used in place of that word.

The data is then converted into a bag of words model where every word is given a unique integer id which is used later for data prediction.

4.2 The Deep Learning RNN Brain Model

The neural network consists of an input layer, several hidden layers and an output layer. The inputs are passed from the input layer to the output layer via the hidden layers where the weights are multiplied with the inputs. The result obtained is backpropagated the entire network, the weights are updated again and the next epoch begins. This is done until the accuracy of the predictions is well over 95 percent. If the number of hidden layers is increased, the chatbot has a chance of understanding the input data better.

4.3 Training and Testing the Chatbot

For the generative chatbot, the dataset has to be very large of at least one million conversations to effectively train the chatbot. The dataset is constantly updated with conversations between people which the model will use to predict answers for input put forward by the user and trained upon previous conversations, based on which the responses to the user are generated. The encoder-decoder component of the LSTM works to analyze and get the prediction vector required for data prediction. The encoder processes the vector of input words, by checking the importance of every word likely to be in the answer and passes this information onto the decoder as encoder state. The decoder takes the output of encoder as the input and makes the prediction using the test prediction vector. The output is then passed onto the output layer to be displayed. This epoch is done a number of times to increase and accuracy and reduce the training error loss.

5. CONCLUSION

The chatbot thus created will solve the problem of the students who live far away to enquire about various details like upcoming events, faculty information, etc. eliminating the need to go to the college specifically to enquire. The users also will have the opportunity to chat with the other chatbot with any conversation the user would like to have as the chatbot talks very similar to a human.

REFERENCES

[1] Kumar Shridhar, May 2017, https://medium.com/botsupply/generative-model-chatbots-e422ab08461e

[2] May 2019, Hidden layers in Neural Networks, https://www.i2tutorials.com/technology/hidden-layers-in-neural-networks/

[3] Synced, September 2017, A brief overview of attention mechanism, https://medium.com/syncedreview/a-brief-overview-of-attention-mechanism-13c578ba9129

[4] Shaojie Jiang, Maarten de Rijke, 2018, Why are sequence-to-sequence Models so dull? Understanding the Low-Diversity problem of Chatbots, publishes in ArXiv, by Cornell University.

[5] Yuening Jia, Attention Mechanism in Machien Translation, 2019, published in Journal of Physics Conference series, 1314:012186, DOI: 10.1088/1742-6596/1314/1/012186.

[6] Ilya Sutskever, Oriol Vinyals, Quoc V. Le, 2014, Sequence to Sequence Learning with Neural Networks, Dec 2014, published in ArXiv, by Cornell University.

[7] Oriol Vinyals, Quoc Le, 2015, A Neural Conversational Model, published in ArXiv, by Cornell University.

[8] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio, 2014, Neural Machine Translation by Jointly Learning to Align and Translate, published in ArXiv, by Cornell University.

[9] Minh-Thang Luong, Hieu Pham, Christopher D. Manning, 2015, Effective Approaches To Attention-

Based Neural Machine Translation, publishes in ArXiv, by Cornell University.

BIOGRAPHY

Ahmed Abdullah is a budding data scientist currently pursuing his Bachelors in Technology. He is a highly analytical individual with strong statistical analysis and research skills and a solid Computer Science background.