

DESIGN AND DEVELOPMENT OF AI-BOT USING  
SEQUENCE TO SEQUENCE LSTM

A Project Report

*Submitted by*

Ms. ASWATHY SANTHOSH

REG NO : TKM20MEAI03

SEMESTER : IV

*In partial fulfillment for the award of the degree of*

MASTER OF TECHNOLOGY

IN

Mechanical Engineering (Artificial Intelligence)

Under the guidance of  
Prof. SUMOD SUNDAR



**Thangal Kunju Musaliar College of Engineering  
Kerala**

JULY 2022

## DECLARATION

I undersigned hereby declare that the project report “**Design and development of AI-Bot using Sequence to Sequence LSTM**”, submitted for partial fulfillment of the requirements for the award of degree of Master of Technology of the APJ Abdul Kalam Technological University, Kerala is a bonafide work done by me under supervision of **Prof. Sumod Sundar**. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

**Place: Kollam**

**Date:**

**Aswathy Santhosh**

Thangal Kunju Musaliar College of Engineering  
Centre for Artificial Intelligence



C E R T I F I C A T E

This is to certify that, this report titled ***DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM*** is a bonafide record of the **project** presented by **ASWATHY SANTHOSH (TKM20MEAI03)**, under our guidance and supervision, in partial fulfillment of the requirements for the award of the degree, **M.Tech in Mechanical Engineering (Artificial Intelligence)** in **APJ Abdul Kalam Technological University** .

Project coordinator & Internal Supervisor

Head of the Department

Prof. Sumod Sundar  
Assistant Professor  
Centre for Artificial Intelligence

Dr. Imthias Ahamed  
Professor & HOD  
Centre for Artificial Intelligence

Internal Examiner

External Examiner

## ACKNOWLEDGEMENT

A successful project is a fruitful culmination of efforts by many people, some directly involved and some others indirectly, by providing support and encouragement. Firstly I would like to thank the almighty for giving me the wisdom and grace for making my project a memorable one. I thank him for steering me to the shore of fulfillment under his protective wings.

I express my sincere gratitude to **Dr. T A Shahul Hameed** , Principal of T.K.M College of Engineering for giving me an opportunity to present my project. I would like to thank **Dr. Imthias Ahamed T P**, Professor and Head of the Department, Centre for Artificial Intelligence, TKM College of Engineering, Kollam, for his constant support and encouragement throughout the project work.

With a profound sense of gratitude, I would like to express my heartfelt thanks to my guide and Project Coordinator, **Prof. Sumod Sundar**, Assistant Professor, Centre for Artificial Intelligence(AI), TKM College of Engineering, Kollam and **Dr. Santhi Natarajan**, Honorary Professor, Centre for Artificial Intelligence, TKM College of Engineering, Kollam for their expert guidance, cooperation and immense encouragement. I also extend my thanks to the entire faculty and staff members of the Centre for AI, TKMCE, who have encouraged me throughout this work.

I also express my thanks to my loving parents and friends, for their support and encouragement in the successful completion of this work.

**Aswathy Santhosh**

## ABSTRACT

There is a rising trend in using conversational agents as virtual assistants and voice interfaces in many business domains and applications. These agents aided users in getting the information they required by responding to their queries through text input, audio input, or both without human intervention. The conventional chatbots could not produce context-based responses to the user inputs as they could not read the text data from both directions. This work mainly focuses on designing and developing the AI-bot using Sequence to Sequence Long Short Term Memory (Seq2Seq LSTM) for the English language on Cornell movie dialogue corpus and a manually created dictionary dataset. The AI-bot using Seq2Seq LSTM, detected acronyms and generated complete relevant answers to the users' queries. Sentiment analysis retrieved meaningful data from text documents, and it helped the agents/bots to communicate with the users based on the context of the conversation and their emotions. Finally, the performance of AI-Bot using Seq2Seq LSTM is evaluated in terms of accuracy, precision, and recall. The model has received an accuracy of 97% and outperformed the RNN model on both datasets. The model has utilized long-range memory dependencies of data. Also, the performance of BERT is found satisfactory in terms of accuracy.

# Contents

<b>1</b>	<b>INTRODUCTION</b>	<b>1</b>
1.1	General Background . . . . .	1
1.2	Objective(s) . . . . .	2
<b>2</b>	<b>LITERATURE SURVEY</b>	<b>3</b>
<b>3</b>	<b>METHODOLOGY</b>	<b>7</b>
3.1	Proposed technique . . . . .	7
3.1.1	Sequence to Sequence(Seq2Seq) LSTM Model . . . . .	7
3.1.2	BERT Model . . . . .	11
<b>4</b>	<b>EXPERIMENTAL ANALYSIS AND RESULTS</b>	<b>14</b>
4.1	Datasets and Environment . . . . .	14
4.2	Data pre-processing steps . . . . .	14
4.3	Experiments on data pre-processing task . . . . .	15
4.4	Results . . . . .	17
<b>5</b>	<b>CONCLUSION</b>	<b>33</b>
<b>6</b>	<b>FUTURE SCOPE</b>	<b>34</b>
	<b>REFERENCES</b>	<b>35</b>
	<b>LIST OF PUBLICATIONS</b>	<b>37</b>

# List of Figures

3.1	Proposed Framework . . . . .	7
3.2	Design of Seq2Seq LSTM . . . . .	8
3.3	Sample Seq2Seq LSTM model . . . . .	9
3.4	LSTM Network . . . . .	10
3.5	Encoder Decoder LSTM Network . . . . .	11
3.6	Sample BERT Question-Answer chatbot . . . . .	11
3.7	BERT Architecture . . . . .	13
4.1	Tokenization . . . . .	15
4.2	Lower casing . . . . .	15
4.3	Stop words removal . . . . .	16
4.4	Stemming . . . . .	17
4.5	Lemmatization . . . . .	18
4.6	Movie lines dataset . . . . .	19
4.7	Movie conversations dataset . . . . .	20
4.8	Dictionary mapping IDs . . . . .	21
4.9	Cleaned list of conversations . . . . .	22
4.10	Extracted questions after dictionary mapping IDs . . . . .	23
4.11	Extracted answers after dictionary mapping IDs . . . . .	24
4.12	Cleaning of text data . . . . .	25
4.13	Questions mapped to integers . . . . .	26
4.14	Answers mapped to integers . . . . .	27
4.15	Final questions dictionary after mapping questions to integers . . . . .	28
4.16	Final answers dictionary after mapping answers to integers . . . . .	29
4.17	Manually created dictionary . . . . .	29
4.18	Building the Seq2Seq LSTM model . . . . .	30
4.19	AI-Bot Interface with RNN . . . . .	30
4.20	AI-Bot Interface with Seq2Seq LSTM . . . . .	31
4.21	AI-Bot Interface using BERT . . . . .	31
4.22	AI-Bot Interface using Seq2Seq LSTM . . . . .	32

# List of Tables

2.1	Review I . . . . .	5
2.2	Review II . . . . .	6
4.1	Performance measures of AI-Bot with different models on Cornell movie dialogue corpus . . . . .	19
4.2	Performance measures of AI-Bot with different models on manually created dictionary dataset . . . . .	20

## ABBREVIATIONS

AI	Artificial Intelligence
BERT	Bidirectional Encoder Representations from Transformers
DL	Deep Learning
IDF	Inverse Document Frequency
LSTM	Long Short-Term Memory
ML	Machine Learning
NLP	Natural Language Processing
RNN	Recurrent Neural Network
Seq2Seq	Sequence to Sequence
SVM	Support Vector Machine
TF	Term Frequency

# Chapter 1

## INTRODUCTION

### 1.1 General Background

With the emergence of Artificial Intelligence(AI), digitization, and increasing use of the Internet and electronic gadgets, people adopted new technologies and applications for interacting and communicating with each other and the market. There is a trend in using conversational agents such as virtual assistants and voice interfaces for answering customer enquiries, assisting the Human Resource(HR) team in interviewing candidates, doing online marketing, assisting in the E-commercial industry and so on. A conversational agent is a software application powered by AI ,and NLP that understands language and communicates with humans. It helps users get the information they need by responding to their queries through text input, audio input, or both without the need for human intervention. It consistently manages customer interactions 24x7 continuously providing good quality of responses to the users' queries.

People stay longer on a website in their native language as it is user-friendly. A multilingual conversational agent allows enterprises and companies to interact with users speaking various regional languages improving engagement and conversations. It is based on the code-switching strategy, a communicative phenomenon denoting a shift from one language to another within the same speech exchange. It helps companies get a broader range of customers as it reduces the language barrier. A conversational agent is well-received and widely used if it can convincingly communicate in a language that is most comfortable to the user. Multilingual agents become a priority as many organizations turn into the global market.

Natural Language Processing(NLP) is an applied AI approach that helps conversational agents to analyze and understand human language[1]. It provides context and meaning to text-based and audio-based user inputs so that the agent outputs the best response. The DL algorithms such as Bidirectional Long Short Term Memory(LSTM), Sequence to Sequence LSTM network and transformer models such as BERT, BART etc., are used to develop agents[1,2,3]. The sentiment analysis task is performed using classification models such as Support Vector Machine(SVM), Naive Bayes(NB) and Decision Tree[4,5]. In this work, a conversational agent named AI-bot is designed and developed using NLP pre-processing techniques and Deep Learning(DL) models such as Seq2Seq LSTM and BERT.

## **1.2 Objective(s)**

- The design and development "AI-Bot" for English language on Cornell movie corpus and a manually created dictionary dataset using Sequence to Sequence Long Short Term Memory (Seq2Seq LSTM).
- The sub-problems of the work are:
  - Context based response generation
  - Generation of complete relevant answers to the users' queries
  - Identification of acronyms and response generation
- Comparing the performance measures of AI-bot using Seq2Seq LSTM with RNN and BERT models.

## Chapter 2

# LITERATURE SURVEY

Wei et al. proposed a Bidirectional LSTM model with multi-polarity orthogonal attention for performing implicit sentiment analysis in Chinese [1]. The multi-polarity attention mechanism provided an attention-based model for finding the differences in sentiment polarities, thereby making it a major attribute for classifying the implicit sentiments with no sentiment words as clues. However, they must identify a proper method for representing implicit external knowledge. Prasad Sahu et al. presented a Lexicon-based approach for sentiment analysis where each word in a sentence is considered to calculate the weight of the opinion words to compute the total polarity of the sentence [2]. They used lexicon resources such as SentiWordNet3.0 and Affin111 for detecting polarity. The sentiment analysis is performed using two classifiers, SVM and Naïve Bayes. However, there is a need to address specific problems such as spam detection, sarcasm detection, conditional statement and anaphora resolution to improve the model's accuracy.

Nuruzzaman et al. implemented a chatbot called IntelliBot in the insurance domain using Natural Language Processing techniques and the LSTM algorithm [3]. It is trained on two datasets, such as the Cornell movie dialogue dataset and a custom-built insurance dataset and achieved more accuracy in response generation as compared to other models such as DeepQA, ChatterBot and RootyAI. In future, they could focus on handling long coherent texts and compare the performance analysis of IntelliBot with other chatbots based on Transformer models. Also, they could research how to leverage reinforcement learning to improve the generated responses' importance, prevent the model from giving appalling responses, and assess its self-learning capabilities. Xu Han et al. presented a computational framework for analysing the performance of interview chatbots[4]. They implemented an assistive chatbot design tool named 'iChatProfile'. It automatically generated a profile of an interview chatbot with estimated performance metrics and provided suggestions on chatbot design. However, they should evaluate the efficiency of the results in different types of interviews. Also, based on recognized themes and the computed performance metric, it could recommend the most appropriate design suggestions for developing chatbots.

Chuan et al. developed a conversational agent in the healthcare domain, including a user-centred classifier [5]. It utilised an active deep learning CNN model for separating complex eligibility criteria into questions that could be quickly replied to by users and information that needed verification by medical experts. It achieved better performance measures as

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

compared to the k-nearest neighbour (KNN) model. It aided users in understanding the eligibility measure of a clinical trial better than a static website. In future work, the model could be trained using various other deep learning methods or hybrid methods for the classification manager and doing experiments with different clinical trials using multiple medical terms. Fast et al. implemented Iris, a conversational agent based on linguistic theory, by executing complex or challenging requests interactively by combining commands through nested conversations [6]. The users have done the task using sklearn in a Jupyter notebook and the Iris conversational interface in random order to nullify learning effects. In future work, they need to examine the accuracy of command classification as the set of commands increases and the number of user examples required to support a new command among a library of thousands of commands.

Touim et al. proposed a chatbot that met learners' requests in a natural language and guided and assisted both the learners and tutors. The model is developed using the LDA Bayesian statistical approach and then classified to get improved semantic responses [7]. In the future, they can use the combination of the LDA approach with the Hawkes process to improve the time factor. Pillai et al. researched the customers' behavioural intention and actual usage (AUE) of artificial intelligence (AI) powered chatbots for hospitality and tourism in India [8]. It is performed by expanding the Technology Adoption Model (TAM) with context-specific variables. Khanuja et al. developed the first dataset for code-mixed Natural Language Inference (NLI), in which both the premises and hypotheses are in code-mixed Hindi-English languages [9]. They used the data from Hindi movies as premises and crowd-sourced hypotheses from Hindi-English bilinguals, which are performed using the mBERT algorithm. The dataset consisted of 400 premises with 2240 hypotheses. They could later annotate the dataset for other linguistic phenomena to solve more complex tasks. Benlahbib et al. proposed a framework to generate a reputation-based system for the mining and fusing opinions expressed in natural languages and user feedback ratings using Machine Learning models such as Support Vector Machine (SVM) and Naïve Bayes classifiers [10]. Also, using various datasets for each domain and language is not practical. Moreover, a filtering phase could be added to the model to decrease the processing time, eventually increasing the model's accuracy.

The advantages and limitations of some recent related works are summarized in tables 2.1 and 2.2.

Table 2.1: Review I

Reference	Technique(s)	Advantages	Disadvantages
[1]	Bidirectional-LSTM	<ul style="list-style-type: none"> <li>• Explicit sentiment analysis is described in this work. The explicit sentiment word in a text or a conversation could be a significant clue for identifying the overall polarity of the text.</li> <li>• Implicit sentiment analysis is performed where an implicit sentiment word in a text indirectly shows the polarity of the conversation.</li> </ul>	<ul style="list-style-type: none"> <li>• External Knowledge Base could be incorporated into the model.</li> <li>• There is a need to explore semantic inference mechanism based implicit sentiment identification methods for identifying the polarities of the text data.</li> </ul>
[2]	Classification techniques: Support Vector Machine (SVM) and Naive Bayes	<ul style="list-style-type: none"> <li>• Negation term is replaced by the negation symbol.</li> <li>• Intensifiers are assigned with polarities such as positive, negative, and neutral.</li> <li>• Punctuations are classified as positive polarity in the text data.</li> </ul>	<ul style="list-style-type: none"> <li>• Certain acronyms and abbreviations need to be handled to calculate the polarity of the text. Examples: Good – Gud or gud, Great – Gr8 or gr8</li> <li>• Sarcasm detection could be included in the work.</li> <li>• Better prediction polarity of comments is needed.</li> </ul>

Table 2.2: Review II

Reference	Technique(s)	Advantages	Disadvantages
[3]	Naïve Bayes, Linear SVM, BERT	<ul style="list-style-type: none"><li>• Classified words into positive and negative polarities in the text.</li><li>• Fusion of opinions and calculation of reputation score are done in this work.</li></ul>	<ul style="list-style-type: none"><li>• Incorporation of filtering phase to neglect irrelevant opinions could be applied to the model.</li><li>• Machine translation can be used.</li></ul>
[4]	RNN, NLP Techniques such as Tokenization, Lemmatization, and POS Tagging	<ul style="list-style-type: none"><li>• The model almost provided relevant answers to the user inputs.</li></ul>	<ul style="list-style-type: none"><li>• Self learning capability of chatbot could be improved by applying hybrid architectures.</li><li>• Coherent text could be handled to get an improved version of the model in the future.</li></ul>

## Chapter 3

# METHODOLOGY

### 3.1 Proposed technique

The AI-Bot is developed using Sequence to Sequence (Seq2Seq) LSTM, and the proposed framework is shown in figure 3.1.

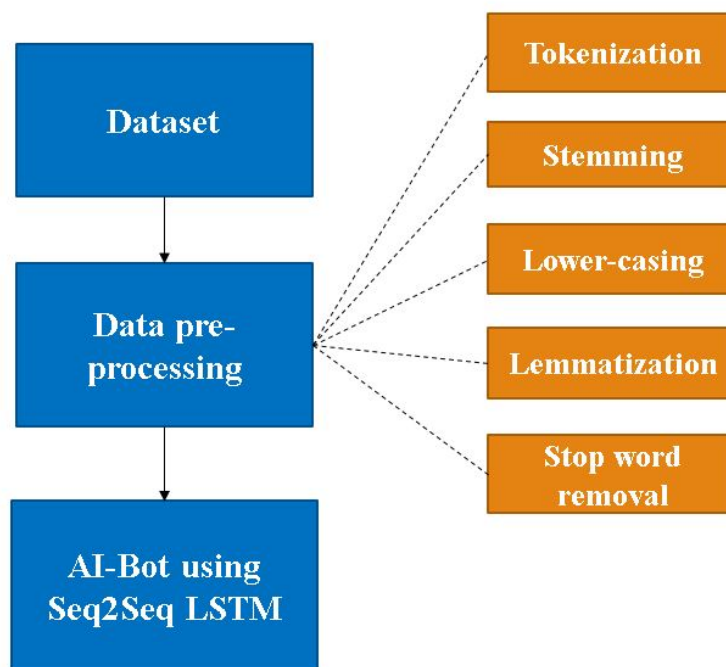


Figure 3.1: Proposed Framework

#### 3.1.1 Sequence to Sequence(Seq2Seq) LSTM Model

The Sequence to Sequence(Seq2Seq) LSTM is used for developing the Deep Learning(DL) part of the AI-bot. It consists of two Long Short Term Memory(LSTM) cells at the:

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

- encoder and
- decoder parts.

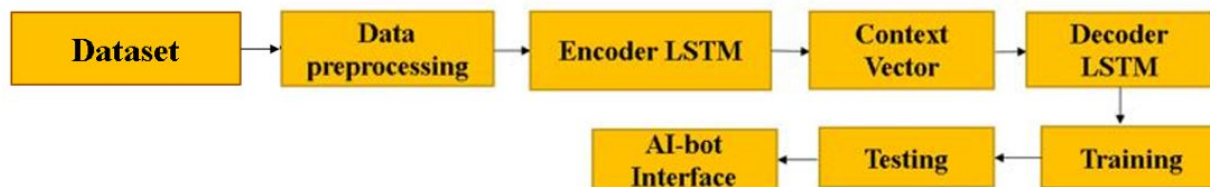


Figure 3.2: Design of Seq2Seq LSTM

The Cornell movie dialogue corpus and the manually created dictionary datasets are fed into the data pre-processing part of the Seq2Seq LSTM model separately. The data pre-processing part is one of the important tasks of the creation of a conversational bot because the raw unstructured datasets are converted to structured forms and is explained in detail in section 4.2. The encoder part reads the input sequence, word by word, and emits a context capturing the semantic summary of the input sequence. Based on context, the decoder generates the output sequence, one word at a time, while looking at the context and the previous word during each time step. The context vectors of the input sequence of words are created using the process of vectorization where each word in a conversation is converted to its corresponding vectorized form of integers. The vectorization method used for creating AI-bot is the Term Frequency-Inverse Document Frequency (TF-IDF) approach. Term frequency is the frequency of a word in a text document or a conversation and is computed as the ratio of the number of times a word occurring in a text document to the total number of words in the document for a specified word. In other words, it is defined as the percentage of the number of times a word ( $x$ ) appears in a specific document ( $y$ ) divided by the total number of words in the text document. Consider the sample conversation for computing the Term frequency of particular word in the text.

$$\text{Term - Frequency}(TF) = \frac{x}{y}$$

Sample conversation: " A language is defined as a structured form of communication where the structure of the language is its grammar and the free parts are its vocabulary. We use various languages for communication purposes and are conveyed through speech, writing, or signs." The term frequency of the word "language" in this document can be calculated as the following. It is calculated in such a manner that all stop words such as a, is, are, the etc., are removed from the document. The term "language" appeared three times in the document and the total number of terms in the document is twenty three and it is shown below.

$$TF(\text{language}) = \frac{3}{23}$$

Inverse Document Frequency (IDF) computes the importance of a particular term in the entire corpus and measures how common a specific term is among all the text documents

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

or conversations in the corpus. It is defined as the logarithmic ratio of the total number of documents to the number of documents with a particular term in them. The following equation shows the ratio of total number of documents( $p$ ) and number of documents with a specific term( $q$ ).

$$IDF = \log\left(\frac{p}{q}\right)$$

The Seq2Seq networks predicted a word given in the user input, and the next word is predicted using the probability of likelihood of that word occurring. The encoder takes the input sequence and aggregates the data in internal state vectors, also known as context vectors (known as the hidden state and cell state vectors in LSTM). The encoder's outputs are discarded, leaving only the internal states. This context vector seeks to incorporate the information for all input elements to enable the decoder to make appropriate predictions. The end result is the neural network which tells out a probability score for the words predicted at the decoder. The neural network will analyze all the terms and it will pick the word with the highest probability. The Seq2Seq model ensures that the word with the highest probability will be given as the output at the decoder part. Figure 3.2 shows the Seq2Seq framework, and figure 3.3 shows Seq2Seq representation.

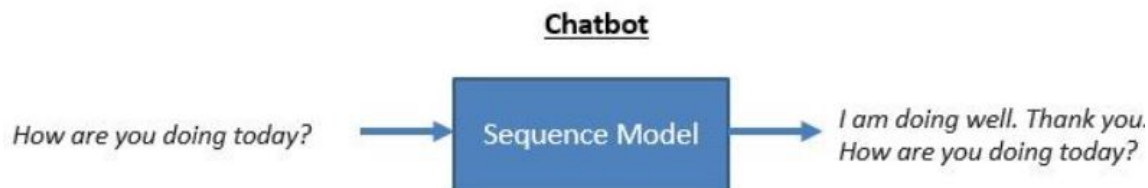


Figure 3.3: Sample Seq2Seq LSTM model

### Long Short Term Memory(LSTM)

LSTM networks are a type of Recurrent Neural Network (RNN) that can learn order dependence in sequence prediction tasks[1]. The LSTM is an enhanced RNN (sequential network) that enables information to be maintained constantly. It can handle vanishing gradient problems occurred with RNNs. It has three gates such as the input gate, forget state and the output state. The first part is called forget gate, the second is known as the input gate, and the last is the output gate. The initial step in an LSTM network cell is to select whether to keep or discard the information from the preceding timestamp, which is done using the forget gate. The activation function,  $f(t)$ , used in this part is the sigmoid function which ranges from 0 to 1. If  $f(t)$  is 0, then the network will forget everything, and if the value of  $f(t)$  is 1, then it will forget nothing. The input gate measures the significance of new data carried by the input. The new data that must be transmitted to the cell state is now a function of a concealed state at timestamp  $t-1$  and input  $x$  at timestamp  $t$ .  $\tan(h)$  is the activation function here. The value of new information will be between -1 and 1 due to the  $\tan(h)$  function. The information is deducted from the cell state if the value of  $N(t)$  is negative and

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

added to the cell state if the value is positive at the current timestamp[1,2]. The output gate turns out that the hidden state is a function of Long term memory ( $C(t)$ ) and the current output. Figure 3.4 shows the LSTM network.

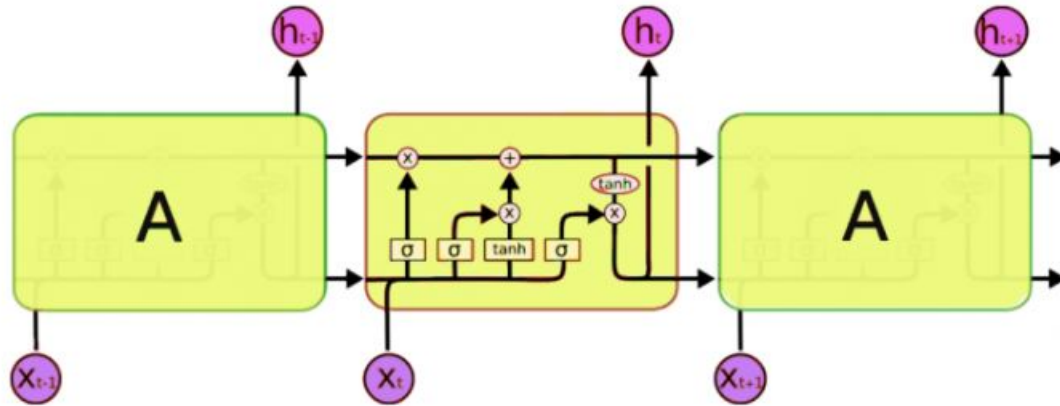


Figure 3.4: LSTM Network

### Encoder and Decoder LSTMs

A stack of many recurrent units (for higher performance, LSTM cells) accepts a single element of the input sequence, accumulates information for that element, and propagates it forward[1]. The input sequence in a question-answering problem is a collection of all words from the query. Each word is represented by  $X(i)$ , where 'i' indicates the word's order. The encoder vector is the model's final hidden state, generated by the encoder. The formula above is used to calculate it. This vector seeks to incorporate all input element information to aid the decoder in making correct predictions[4].

A set of recurrent units, each of which predicts an output,  $y(t)$  at a given time step "t". Each recurrent unit takes a secret state from the preceding unit and outputs its hidden state. The output sequence in the question-answering problem is a collection of all words or terms from the answer. Each word is represented as  $Y(i)$ , where "i" denotes the word's order. The outputs are computed by combining the hidden state at the current time step with the weight  $W(s)$ . Softmax is used to generate a probability vector that will help predict the final result[6,7]. The Encoder-Decoder LSTM network is shown in figure 3.5.

### Workflow of the Seq2Seq LSTM model

The steps done in the training and testing phases of the model are as the following:

- Importing the libraries
- Data pre-processing steps
- Building the encoder part

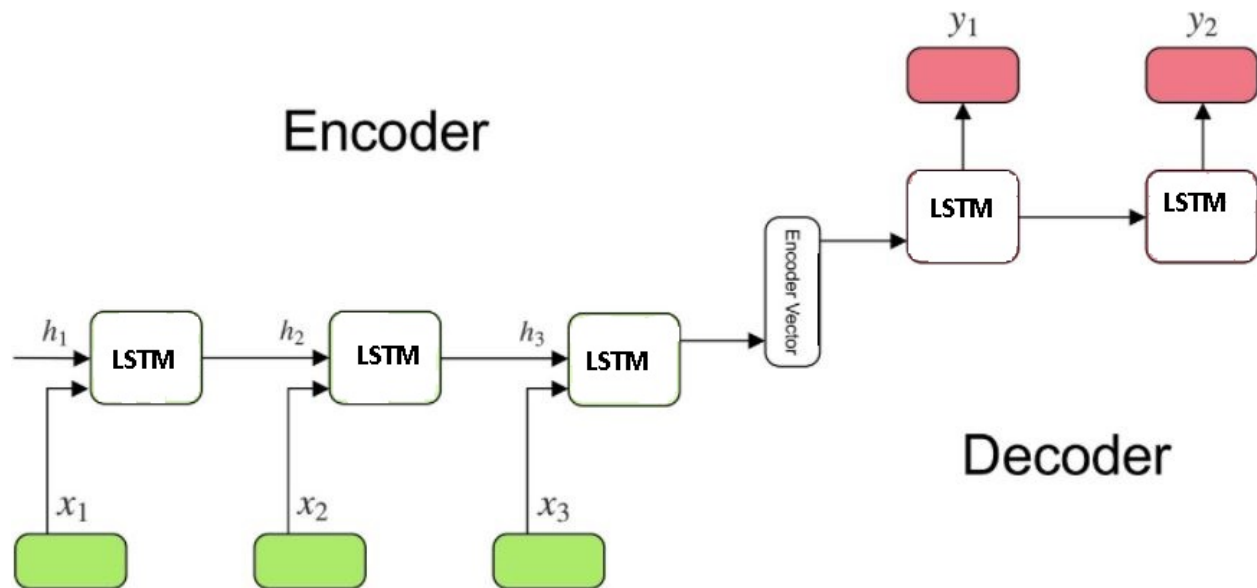


Figure 3.5: Encoder Decoder LSTM Network

- Training and testing the encoder LSTM
- Building the decoder LSTM
- Training and testing decoder part
- Developing the Seq2Seq LSTM AI-Bot

### 3.1.2 BERT Model

Bidirectional Encoder Representations from Transformers (BERT) is a Machine Learning (ML) model developed especially for natural language processing-based applications by Google [9]. The BERT has two models:

- BERT-BASE: 12 encoders with 12 bidirectional self-attention heads
- BERT-LARGE: 24 encoders with 16 bidirectional self-attention heads



Figure 3.6: Sample BERT Question-Answer chatbot

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

Figure 3.6 shows the sample BERT Question-Answer chatbot. These two models are pre-trained from unlabeled data extracted from the Books Corpus with 800 million words and English Wikipedia with 2,500 million words. BERT follows a semi-supervised learning approach and enables one to understand the language. It has language processing capabilities that are used to strengthen other models. The AI-bot is developed using the BERT algorithm as a generative Question-Answering conversational bot. Sentiment analysis and context-based conversations are generated using the AI-bot. BERT was designed and developed to assist computers or other smart devices in understanding the meaning of the ambiguous text in a conversation. It was pre-trained using text from Wikipedia. Also, it could be fine-tuned with question-answer datasets such as the Cornell movie dialogue corpus and a manually created dictionary dataset. BERT is a transformer-based ML algorithm for the NLP pre-training process designed and developed by Google [9]. The BERT algorithm utilizes a bidirectional training approach as it reads a sentence in a text or a conversation from both directions to understand the context and meaning of the sentence. The BERT architecture is shown in figure 3.7. The Cornell movie dialogue corpus and the manually created dictionary datasets are used for building the model. The initial step is the data pre-processing part and the cleaned, structured datasets are fed into the BERT model. Vectorization of the input text is done using the TF-IDF approach where each term is converted to its corresponding vector and is fed into the transformer to predict the answer corresponding to the specific text input.

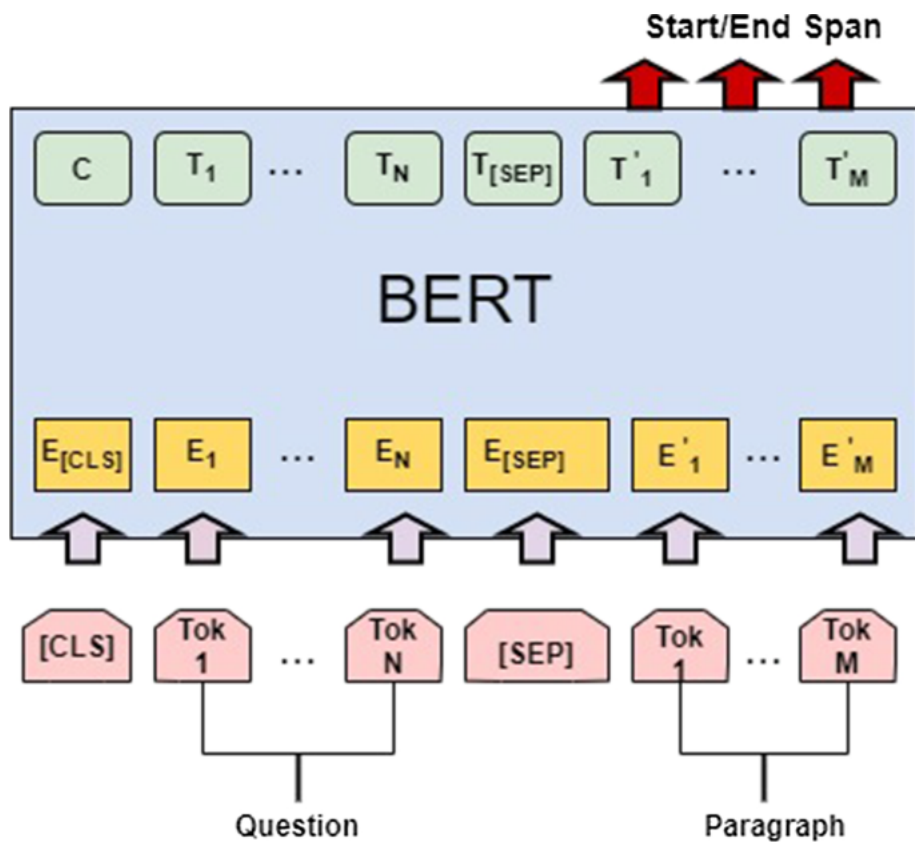


Figure 3.7: BERT Architecture

## Chapter 4

# EXPERIMENTAL ANALYSIS AND RESULTS

### 4.1 Datasets and Environment

The datasets used in this work are the Cornell Movie Dialogue Corpus and a manually created dictionary. The Cornell Movie Dialogue Corpus consists of Two hundred twenty thousand five hundred seventy-nine conversational exchanges between 10,292 pairs of movie characters involve 9,035 characters from 617 movies. There are 304,713 utterances in total. The manually created dictionary dataset consists of general queries and corresponding answers[4]. A virtual environment named “Chabot” is created in the Anaconda platform and the libraries used are Numpy, Tensorflow, Re, and Time. The NumPy library works with arrays, TensorFlow deals with the Deep Learning(DL) techniques, and Re library cleans the text and makes it easy for the chatbot to learn. The manually created dictionary dataset is prepared manually by incorporating generative question-answer patterns from various domains such as countries, history, politics, geography, culture, and healthcare. The abbreviations or the acronyms of various words are incorporated in the datasets so that the AI-bot identifies these and generates responses by understanding the meaning and context of the user text inputs.

### 4.2 Data pre-processing steps

The detailed framework of the AI-Bot is shown in figure 4.1. The various data pre-processing steps in the development of AI-Bot are described in this section.

- Tokenization
- Lower casing
- Stop words removal
- Stemming
- Lemmatization

Tokenization is the initial step in the data pre-processing task, and it is the process of splitting or breaking the complete sentence into various chunks of data. Figure 4.2 shows

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

the Tokenization process. The lower casing is the process of converting a word to lower case. For example: Consider the word "CAT", which is converted into "cat", as shown in figure 4.3. Stop words are commonly used in documents and conversations, such as a, an, the, etc. The stop words do not contribute any importance as they do not help in differentiating two documents. Stop words are removed from the text, which plays a significant role in the data cleaning part of the documents. Stop word removal is shown in figure 4.4. Stemming is the process of transforming or converting a word to its root or base form and is shown in figure 4.5. Lemmatization is the process of reducing a word to its dictionary form called root form, and the root word is called a 'lemma'. Lemmatization is preferred over stemming because it performs morphological analysis of the words and is shown in figure 4.6.

```
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize

sentence = "I am preparing for university examination"

words = word_tokenize(sentence)
print(words)

['I', 'am', 'preparing', 'for', 'university', 'examination']
```

Figure 4.1: Tokenization

```
sentence = "I AM Preparing For University EXAMINATION."
sentence = sentence.lower()
print(sentence)

i am preparing for university examination.
```

Figure 4.2: Lower casing

### 4.3 Experiments on data pre-processing task

The data pre-processing steps are described in this section. The movie lines.txt contains some extracts of scripts taken from the movie and consists of five columns. The first column is the ID of the line said by some characters in a movie. The second column represents the user

```
nlTK.download('stopwords')
from nlTK.corpus import stopwords
from nlTK.tokenize import word_tokenize

sentence = "I am going to the church"

stop_words = set(stopwords.words('english'))
word_tokens = word_tokenize(sentence)

filtered_sentence = [w for w in word_tokens if not w in stop_words]
print(filtered_sentence)

['I', 'going', 'church']
```

Figure 4.3: Stop words removal

character(u), the third column consists of movie numbers(m), the fourth column contains the name of the characters(metadata/description data), and the fifth column includes the conversations in the movies. Importing the datasets is the initial step of the data pre-processing, and the open() function is used to load the datasets such as the movie lines.txt and movie conversation.txt into the model. Identifiers are given in the conversations. Figure 4.7 shows the movie lines dataset, and the movie conversations dataset is shown in figure 4.8.

The second step is to create a python dictionary that maps each line IDE. This dictionary consists of the line IDs and the corresponding conversational line w.r.t each ID, which is shown in figure 4.9. The dataset is split so that the key identifier or ID is mapped with the last element. The next step is the creation of a list of all conversations. The split function and replace function are used for doing this step. The text inputs are cleaned by removing all irrelevant texts from the datasets. It is shown in figure 4.10. The fourth step is extracting the questions and answers separately, shown in figures 4.11 and 4.12. The answers to index "i" should be the answer to the question "i-1", iterating the conversations IDs. Cleaning the texts is the next step in the data pre-processing task, shown in figure 4.13. A dictionary is further created to remove the words that count below a certain threshold. Here, the threshold is set as a count of 20 or 5% of the total number of words' occurrences. In the next step, create a dictionary that maps question words to a unique integer shown in figure 4.14. Similarly, another dictionary which maps answer words to a unique integer is also created, which is shown in figure 4.15.

All sequences in the two dictionaries should have the same length. Therefore the tokens such as padding,  $\text{PAD}$ , End of String,  $\text{EOS}$  and Start of String,  $\text{SOS}$  are added to these dictionaries. Also, there is a need to create an inverse mapping dictionary. Answer words to integers are reversed to get the integers to answer words dictionary. The final step is

```
import nltk
from nltk.stem import PorterStemmer
ps = PorterStemmer()

sentence = "I watched a film yesterday and meted my friend at the theatre."

for word in sentence.split():
    print(ps.stem(word))
```

```
i
watch
a
film
yesterday
and
meet
my
friend
at
the
theatre.
```

Figure 4.4: Stemming

mapping all questions and answers into integers, which are shown in figures 4.16 and 4.17.

The manually created dictionary dataset is shown in figure 4.17. The pre-processing of the datasets is performed, and the final data are fed into the training and testing phase of the seq2seq model. The text inputs in the encoder are converted to a vector form sent to the middle part of the model, named context vector. The decoder decodes the context vector and reverses the process of the encoder. It converts all the vectors into text which the end-user can get at the output phase. The result is the neural network which tells the probability score for its options. So, it will assign probabilities to the various words it can pick from the dataset. It analyzed all the words and picked the word with the highest probability. Twenty thousand predictions are possible at each output stage of the model.

## 4.4 Results

The Seq2Seq model mapped a fixed-length input with a fixed-length output where the length of input text and output may differ. The model would predict the most appropriate response at the user side by performing the back propagation technique, thereby making sure that the word with the highest probability is predicted at the output stage. It updates the weights and the parameters of the network so that the output word having the highest probab-

```
import nltk
nltk.download('wordnet')
nltk.download('omw-1.4')
from nltk.stem import WordNetLemmatizer

lemmatizer = WordNetLemmatizer()

print(lemmatizer.lemmatize("Machine", pos='n'))
# pos: parts of speech tag, verb
print(lemmatizer.lemmatize("caring", pos='v'))
print(lemmatizer.lemmatize("knowing", pos='v'))
print(lemmatizer.lemmatize("thinking", pos='v'))
print(lemmatizer.lemmatize("sang", pos='v'))
print(lemmatizer.lemmatize("caught", pos='v'))
```

Machine  
care  
know  
think  
sing  
catch

Figure 4.5: Lemmatization

ity is observed on the user side[8]. The weights are trained using the back-propagation and stochastic gradient descent algorithm. Through training, the decoder LSTM will know which elements need more attention. The output phase of the model using the Cornell movie dialogue corpus is shown in figure 4.18. The total number of batches is 2700 with 50 epochs. Each epoch is trained by using 100 batches up to 50 epochs. The word ambiguities in a conversation are identified by generating the term with the highest probability as responses to the user inputs. The user will pass a general query to the AI-bot, and it will respond to it with the most appropriate matching response. Figure 4.19 shows the model interface using RNN.

The AI-bot interfaces created using the Seq2Seq LSTM on the manually created dictionary dataset are shown in figures 4.20 and 4.22 . Figure 4.21 shows the AI-bot interface using BERT. Table 4.1 shows the comparison of performance measures of AI-bot using different models on the Cornell movie dialogue corpus. The AI-Bot using Seq2Seq LSTM on the Cornell movie dialogue corpus obtained an accuracy of 97%, precision of 0.97, recall of 0.98 and outperformed the RNN model in terms of these measures. Similarly, the AI-Bot using Seq2Seq LSTM on the manually created dictionary dataset achieved better performance

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

Index	Type	Size	
0	str	1	L1045 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ They do not!
1	str	1	L1044 +++\$+++ u2 +++\$+++ m0 +++\$+++ CAMERON +++\$+++ They do to!
2	str	1	L985 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ I hope so.
3	str	1	L984 +++\$+++ u2 +++\$+++ m0 +++\$+++ CAMERON +++\$+++ She okay?
4	str	1	L925 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ Let's go.
5	str	1	L924 +++\$+++ u2 +++\$+++ m0 +++\$+++ CAMERON +++\$+++ Wow
6	str	1	L872 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ Okay -- you're gonna ...
7	str	1	L871 +++\$+++ u2 +++\$+++ m0 +++\$+++ CAMERON +++\$+++ No
8	str	1	L870 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ I'm kidding. You kn ...
9	str	1	L869 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ Like my fear of wear ...
10	str	1	L868 +++\$+++ u2 +++\$+++ m0 +++\$+++ CAMERON +++\$+++ The "real you".
11	str	1	L867 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ What good stuff?
12	str	1	L866 +++\$+++ u2 +++\$+++ m0 +++\$+++ CAMERON +++\$+++ I figured you'd get ...
13	str	1	L865 +++\$+++ u2 +++\$+++ m0 +++\$+++ CAMERON +++\$+++ Thank God! If I ha ...
14	str	1	L864 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ Me. This endless .. ...
15	str	1	L863 +++\$+++ u2 +++\$+++ m0 +++\$+++ CAMERON +++\$+++ What crap?
16	str	1	L862 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ do you listen to thi ...
17	str	1	L861 +++\$+++ u2 +++\$+++ m0 +++\$+++ CAMERON +++\$+++ No...
18	str	1	L860 +++\$+++ u0 +++\$+++ m0 +++\$+++ BIANCA +++\$+++ Then Guillermo says, ...

Figure 4.6: Movie lines dataset

measures with an accuracy of 97%, precision and recall of 0.98 and 0.97, respectively, than the RNN model. Also, the performance of BERT is found satisfactory in terms of accuracy, precision and recall. Table 4.2 shows the comparison of performance measures of AI-bot using different models on the manually created dictionary dataset.

Table 4.1: Performance measures of AI-Bot with different models on Cornell movie dialogue corpus

Model	Accuracy	Precision	Recall
RNN	96%	0.95	0.95
Seq2Seq LSTM	97%	0.97	0.98
BERT	97%	0.98	0.98

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

Index	Type	Size	
0	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L194', 'L195', 'L196', 'L197']
1	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L198', 'L199']
2	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L200', 'L201', 'L202', 'L203']
3	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L204', 'L205', 'L206']
4	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L207', 'L208']
5	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L271', 'L272', 'L273', 'L274', 'L27 ...
6	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L276', 'L277']
7	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L280', 'L281']
8	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L363', 'L364']
9	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L365', 'L366']
10	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L367', 'L368']
11	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L401', 'L402', 'L403']
12	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L404', 'L405', 'L406', 'L407']
13	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L575', 'L576']
14	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L577', 'L578']
15	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L662', 'L663']
16	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L693', 'L694', 'L695']
17	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L696', 'L697', 'L698', 'L699']
18	str	1	u0 +++\$+++ u2 +++\$+++ m0 +++\$+++ ['L860', 'L861']

Figure 4.7: Movie conversations dataset

Table 4.2: Performance measures of AI-Bot with different models on manually created dictionary dataset

Model	Accuracy	Precision	Recall
RNN	95%	0.95	0.94
Seq2Seq LSTM	97%	0.98	0.97
BERT	97%	0.98	0.98

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

Key	Type	Size	
L1000	str	1	Oh, Christ. Don't tell me you've changed your mind. I already sent ' ...
L10000	str	1	Oh... chamber runs. Uh huh, that's good. Well, hey... you guys know ...
L100000	str	1	No.
L100001	str	1	Then why did you go see Mr. Koehler in the first place?
L100002	str	1	Because he called me.
L100003	str	1	Hi, Joe.
L100004	str	1	Frank -- what are you doing here?
L100005	str	1	I need to ask you a question. What made you get in touch with Joanne ...
L100006	str	1	You know I can't talk about that.
L100007	str	1	I'm not asking for names or specifics. I just want to know what promp ...
L100008	str	1	Sorry.
L10001	str	1	Those guys ain't so tough. I fought plenty of guys tougher'n them.
L100011	str	1	Hello?
L100012	str	1	Frank -- It's Rebecca. I need to see you right away. I've got the ta ...
L100016	str	1	You killed him. You killed him -- and I got you off.
L100017	str	1	That's crazy.

Figure 4.8: Dictionary mapping IDs

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

Index	Type	Size	
0	list	4	['L194', 'L195', 'L196', 'L197']
1	list	2	['L198', 'L199']
2	list	4	['L200', 'L201', 'L202', 'L203']
3	list	3	['L204', 'L205', 'L206']
4	list	2	['L207', 'L208']
5	list	5	['L271', 'L272', 'L273', 'L274', 'L275']
6	list	2	['L276', 'L277']
7	list	2	['L280', 'L281']
8	list	2	['L363', 'L364']
9	list	2	['L365', 'L366']
10	list	2	['L367', 'L368']
11	list	3	['L401', 'L402', 'L403']
12	list	4	['L404', 'L405', 'L406', 'L407']
13	list	2	['L575', 'L576']
14	list	2	['L577', 'L578']
15	list	2	['L662', 'L663']
16	list	3	['L693', 'L694', 'L695']
17	list	4	['L696', 'L697', 'L698', 'L699']
18	list	2	['L860', 'L861']
19	list	4	['L862', 'L863', 'L864', 'L865']
20	list	4	['L866', 'L867', 'L868', 'L869']
21	list	3	['L870', 'L871', 'L872']

Figure 4.9: Cleaned list of conversations

Index	Type	Size	
0	str	1	Can we make this quick? Roxanne Korrine and Andrew Barrett are having ...
1	str	1	Well, I thought we'd start with pronunciation, if that's okay with you ...
2	str	1	Not the hacking and gagging and spitting part. Please.
3	str	1	You're asking me out. That's so cute. What's your name again?
4	str	1	No, no, it's my fault -- we didn't have a proper introduction ---
5	str	1	Cameron.
6	str	1	The thing is, Cameron -- I'm at the mercy of a particularly hideous br ...
7	str	1	Why?
8	str	1	Unsolved mystery. She used to be really popular when she started high ...
9	str	1	Gosh, if only we could find Kat a boyfriend...
10	str	1	C'esc ma tete. This is my head

Figure 4.10: Extracted questions after dictionary mapping IDs

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

Index	Type	Size	
0	str	1	Well, I thought we'd start with pronunciation, if that's okay with you ...
1	str	1	Not the hacking and gagging and spitting part. Please.
2	str	1	Okay... then how 'bout we try out some French cuisine. Saturday? Nig ...
3	str	1	Forget it.
4	str	1	Cameron.
5	str	1	The thing is, Cameron -- I'm at the mercy of a particularly hideous br ...
6	str	1	Seems like she could get a date easy enough...
7	str	1	Unsolved mystery. She used to be really popular when she started high ...
8	str	1	That's a shame.
9	str	1	Let me see what I can do.
10	str	1	Right. See? You're ready for the quiz.
11	str	1	I don't want to know how to say that though. I want to know useful th ...
12	str	1	That's because it's such a nice one.
13	str	1	Forget French.

Figure 4.11: Extracted answers after dictionary mapping IDs

```
text = re.sub(r"i'm", "i am", text)
text = re.sub(r"he's", "he is", text)
text = re.sub(r"she's", "she is", text)
text = re.sub(r"that's", "that is", text)
text = re.sub(r"let's", "let us", text)
text = re.sub(r"it's", "it is", text)
text = re.sub(r"what's", "what is", text)
text = re.sub(r"where's", "where is", text)
text = re.sub(r"\ll", " will", text)
text = re.sub(r"\ve", " have", text)
text = re.sub(r"\re", " are", text)
text = re.sub(r"\d", " would", text)
text = re.sub(r"won't", "will not", text)
text = re.sub(r"can't", "cannot", text)
text = re.sub(r"don't", "do not", text)
text = re.sub(r"didn't", "did not", text)
text = re.sub(r"wasn't", "was not", text)
text = re.sub(r"[-()\"#/@;:<>{}+=~|.?,]", "", text)
```

Figure 4.12: Cleaning of text data

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

Key	Type	Size	
accident	int	1	8770
accident!	int	1	5500
accidentally	int	1	7008
accidents	int	1	3231
accompany	int	1	7866
accomplish	int	1	196
accomplished	int	1	8714
according	int	1	8161
account	int	1	4619
accountant	int	1	825
accounting	int	1	5142
accounts	int	1	3704
accurate	int	1	4580
accuse	int	1	7131
accused	int	1	5912
accusing	int	1	7410
ace	int	1	4900
achieve	int	1	2494
acid	int	1	679
acknowledge	int	1	2825

Figure 4.13: Questions mapped to integers

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

Key	Type	Size	
captain	int	1	7328
captain's	int	1	7982
capture	int	1	6866
captured	int	1	6458
car	int	1	4724
carl	int	1	202
car's	int	1	6271
carbon	int	1	3342
card	int	1	3290
cardiac	int	1	3955
cards	int	1	1456
care	int	1	4123
carel	int	1	3590
cared	int	1	5659
career	int	1	985
careful	int	1	4741
carefully	int	1	3804
careless	int	1	5520
cares	int	1	7129
cargo	int	1	864
caring	int	1	6348
carl	int	1	4074
carl's	int	1	5807
carla	int	1	4648
carlos	int	1	6916

Figure 4.14: Answers mapped to integers

DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

Index	Type	Size	
0	list	22	[629, 2854, 6140, 7058, 151, 8821, 8821, 5123, 7323, 8821, ...]
1	list	14	[8801, 3222, 1056, 2854, 263, 1634, 1996, 8821, 2217, 5110, ...]
2	list	9	[3795, 8159, 8821, 5123, 8821, 5123, 8821, 3340, 1073]
3	list	14	[4835, 4415, 3507, 3520, 714, 5110, 5265, 603, 2787, 344, ...]
4	list	13	[5096, 5096, 8612, 5265, 3659, 6449, 2854, 4695, 3795, 4404, ...]
5	list	1	[4915]
6	list	24	[8159, 1793, 5265, 4915, 3222, 3317, 4529, 8159, 8049, 8311, ...]
7	list	1	[5065]
8	list	25	[8821, 3817, 4992, 1082, 1886, 587, 4521, 8597, 8084, 4992, ...]
9	list	9	[3227, 2217, 5382, 2854, 4639, 5995, 1235, 2393, 2073]
10	list	7	[8821, 2141, 8821, 7058, 5265, 3659, 4977]
11	list	8	[1065, 6422, 4835, 4415, 8752, 186, 8159, 8821]
12	list	45	[3222, 2720, 3795, 8207, 1886, 3108, 3468, 1886, 1986, 5110, ...]
13	list	9	[5110, 5265, 4876, 8612, 5265, 5186, 2393, 7495, 2747]
14	list	11	[3468, 5265, 7752, 2921, 5995, 8159, 8821, 2393, 8117, 3667, ...]

Figure 4.15: Final questions dictionary after mapping questions to integers

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

Index	Type	Size	
0	list	16	[8801, 3222, 1056, 2854, 263, 1634, 1996, 8821, 2217, 5110, ...]
1	list	11	[3795, 8159, 8821, 5123, 8821, 5123, 8821, 3340, 1073, 8820, ...]
2	list	14	[658, 1057, 3468, 5915, 2854, 4341, 714, 3127, 5128, 8821, ...]
3	list	4	[4149, 8612, 8820, 8820]
4	list	3	[4915, 8820, 8820]
5	list	26	[8159, 1793, 5265, 4915, 3222, 3317, 4529, 8159, 8049, 8311, ...]
6	list	11	[4945, 1705, 4992, 4639, 578, 2393, 8117, 2578, 1301, 8820, ...]
7	list	27	[8821, 3817, 4992, 1002, 1886, 587, 4521, 8597, 8084, 4992, ...]
8	list	6	[5110, 5265, 2393, 6672, 8820, 8820]
9	list	9	[7340, 3520, 6422, 344, 3222, 629, 2720, 8820, 8820]
10	list	10	[1065, 6422, 4835, 4415, 8752, 186, 8159, 8821, 8820, 8820]
11	list	47	[3222, 2720, 3795, 8207, 1886, 3108, 3468, 1886, 1986, 5110, ...]
12	list	11	[5110, 5265, 4876, 8612, 5265, 5186, 2393, 7495, 2747, 8820, ...]

Figure 4.16: Final answers dictionary after mapping answers to integers

```

{"tag": "greeting",
 "patterns": ["Hello", "How are you?", "Hi there", "Hi", "Whats up"],
 "responses": ["Hey", "Hello", "How are you doing?", "Greetings!", "How are you?","I am fine , Thank you"],
},
{"tag": "service",
 "patterns": ["Hello, please help me", "Good", "I am fine", "Help me AI Bot", "What are your services?"],
 "responses": ["Hey, how may I help you?", "Awesome", "Sure,Its my pleasure to help you", "I am here to guide you finding answers t
},
{"tag": "age",
 "patterns": ["how old are you?", "when is your birthday?", "when was you born?","what is your age"],
 "responses": [ "I was created in 2021 by my developer" ]
},
{"tag": "Technology",
 "patterns": ["What is ML", "Tell me something about Machine learning in Artificial Intelligence?", "Machine learning"],
 "responses":["Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically
},
{"tag": "Technology and development",
 "patterns": ["What is AI ", "Tell me something Artificial Intelligence?", "What is Artificial Intelligence (AI)?"],
 "responses": [" Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines c
},
{"tag": "date",
 "patterns": ["what are you doing this weekend?",
to hang out some time?", "what are your plans for this week"],
 "responses": ["I am available all week", "I don't have any plans", "I am not busy"]

```

Figure 4.17: Manually created dictionary

# DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

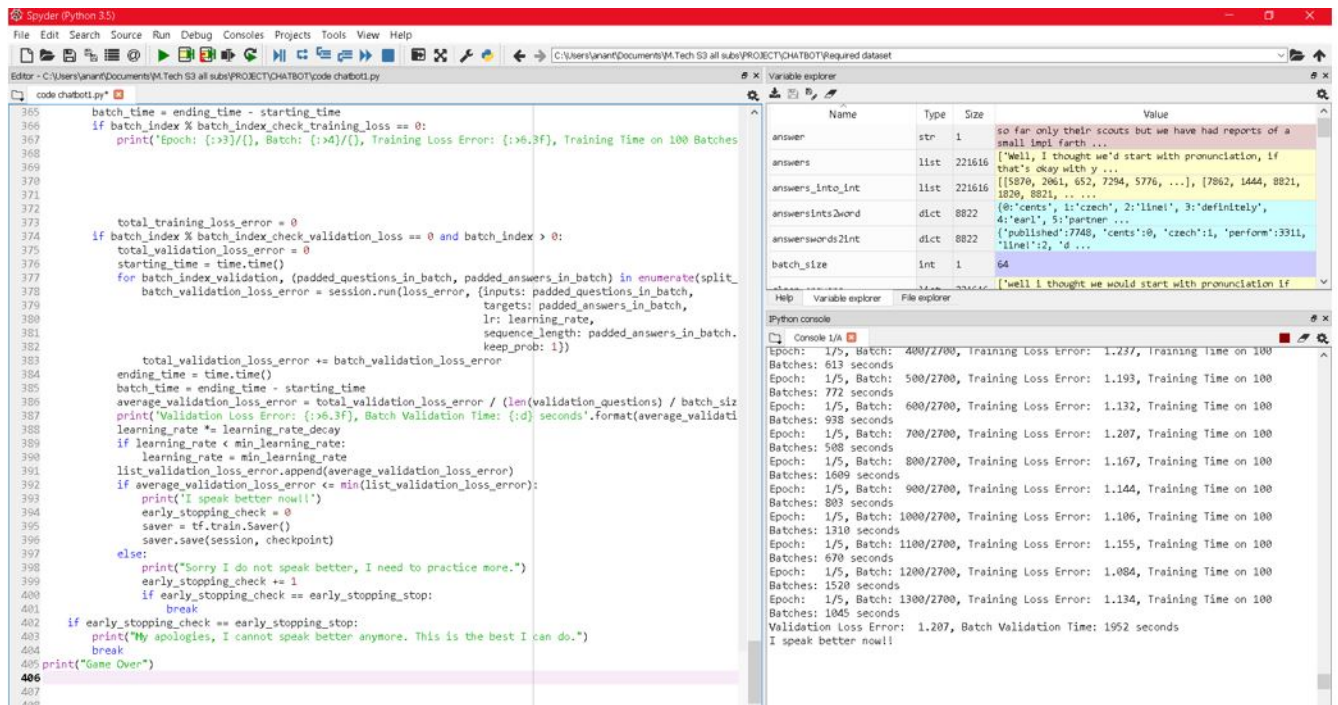


Figure 4.18: Building the Seq2Seq LSTM model

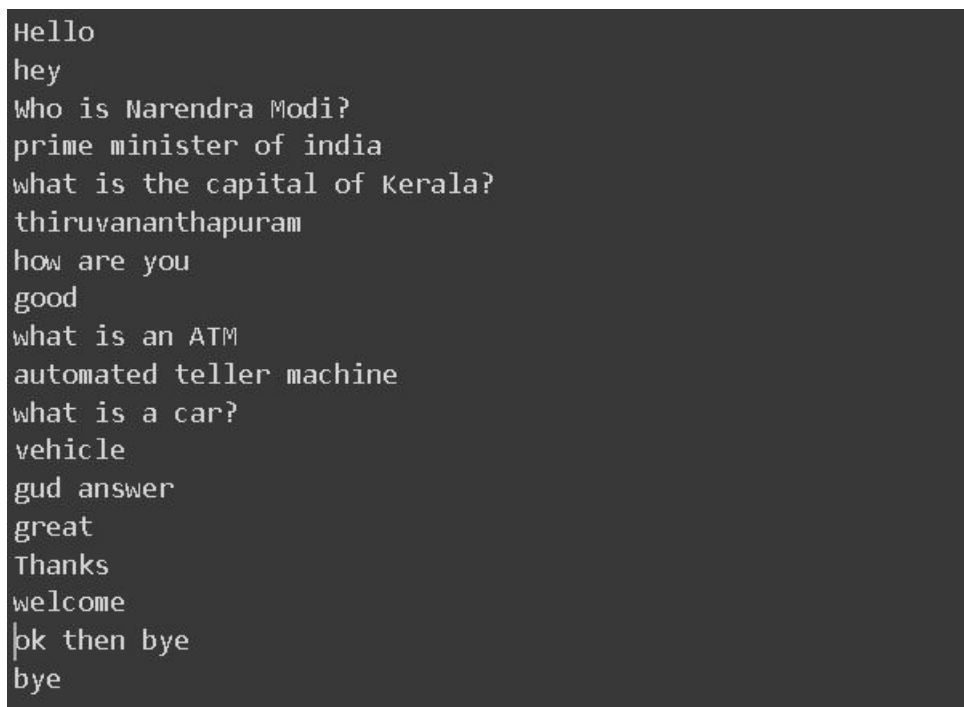


Figure 4.19: AI-Bot Interface with RNN

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

```
... hello
How are you doing?
good
Awesome
who are you
I am AI Bot, how can I help you?
what is ai
Artificial intelligence (AI) is the ability of a computer or a robot controlled by a computer to do tasks that are usually done by humans b
who is the president of india
President of India is Ram Nath Kovind
who is narendra modi
Narendra modi
prime minister of india is
Prime minister of India is Narendra Damodardas Modi
what is deep learning
Deep learning is a subset of machine learning where artificial neural networks, algorithms inspired by the human brain, learn from large am
what is natural language processing
Natural language processing (NLP) is a branch of artificial intelligence within computer science that focuses on helping computers to under
who is the chief minister of kerala
Chief Minister of Kerala is Pinarayi Vijayan
whats your weekend plan
I don't have any plans
good
Sure,Its my pleasure to help you
```

Figure 4.20: AI-Bot Interface with Seq2Seq LSTM

```
Hello AI-bot
Hey Greetings from AI-bot. What can I do for you?
Who are you?
I am AI-bot, an artificial intelligence conversational bot serving as
a Question-Answering agent.
Okay!
Great
Could you help me?
Yes,what can I do for you?
What do you mean by a chatbot?
A chatbot is a type of software that can help customers by
automating conversations and interact with them through messaging
platforms.
Good
Thank you
Who is Narendra Modi?
Narendra Damodardas Modi is an Indian politician serving as the 14th
and current prime minister of India since 2014. Modi was the chief
minister of Gujarat from 2001 to 2014 and is the Member of
Parliament from Varanasi.
Okay
Great
Thanks for sharing
It is my pleasure to share information with you. How can I help you?
What is deep learning?
Deep learning (also known as deep structured learning) is part of a
broader family of machine learning methods based on artificial
neural networks with representation learning.
okay thank you
You are welcome
bye
Thanks for your interaction.Bye
```

Figure 4.21: AI-Bot Interface using BERT

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

```
Enter question : WHICH IS YOUR FAVORITE MOVIE
WARNING:tensorflow:Model was constructed with shape (None, 74) for input KerasTensor(type_spec=TensorSpec(shape=(None, 74), dtype=tf.float32, name='input')), dtype=tf.float32
my favorite story has 2001 end
Enter question : WHICH IS FAVORITE STORY
it's a book story by hans moravec end
Enter question : WHO IS HANS MORAVEC
it is a book of robots end
Enter question : WHAT IS A ROBOT
a computer is a program that attempts to simulate the conversation or chat of a human being a lot of a high castle end
Enter question : WHAT IS A COMPUTER
a computer is an electronic device which takes information in digital form and performs a series of operations based on predetermined instructions end
Enter question : WHAT IS AN ELECTRONIC DEVICE
it is a reference to the illuminatus trilogy that is the number of a wave from any common point of two times ten are you end
Enter question : WHAT IS A WAVE
ideally it is a representative of the people end
Enter question : WHO ARE PEOPLE
my favorite story is 2001 end
Enter question : WHAT IS 2001 END
it is the scientific study of bacteria and diseases caused by them end
```

Figure 4.22: AI-Bot Interface using Seq2Seq LSTM

## Chapter 5

# CONCLUSION

The main focus of the work is the design and development of the AI-bot for the English language using the Cornell movie dialogue corpus and a manually created dictionary dataset. The AI-Bot is created using the Seq2Seq LSTM and compared with the RNN and BERT models in this work. The datasets are cleaned using the pre-processing steps such as tokenization, stemming, lemmatization, stop words removal, punctuation cleaning, and lower casing of texts. The pre-processed datasets are then passed separately to the model building stage, and these are trained and tested using the Seq2Seq LSTM in Spyder IDE in the Anaconda platform. The Seq2Seq LSTM model consisted of an encoder and a decoder, which took the user inputs and eventually generated responses to the text inputs. The model obtained meaningful responses to the text inputs with both datasets. The AI-bot generated complete relevant answers to the questions passed by the users and also detected the acronyms in the conversation.

## Chapter 6

# FUTURE SCOPE

There is a scope for applying advanced NLP techniques for better prediction of the polarity of the reviews. The problems such as spam detection, sarcasm detection, conditional statement, and anaphora resolution can also be considered to improve the model's performance. Exploring and analysing how to handle long coherent text is necessary. A multilingual conversational agent allows enterprises and companies to interact with users speaking various regional languages improving engagement and conversations. It is based on the code-switching strategy, a communicative phenomenon denoting a shift from one language to another within the same speech exchange [13]. Also, there is a need to explore the semantic inference mechanism-based implicit sentiment identification methods for achieving better performance measures. Emotion detection techniques enhance the performance of the conversational agent as it will be able to understand and generate responses w.r.t the users' emotions. Transformer-based models such as mBERT, RoBERTa, BART, etc., could be used in the future to generate better responses. Also, there is a scope of incorporating the images of products, people, organizations, logos, etc., into the AI-bot so that it effectively identifies the images given by the users to it and produces relevant responses [14]. Furthermore, we can develop the AI-bot to converse in different regional languages training using various language datasets, and we could use the indic NLP datasets for building the bot [15].

# References

- [1] Wei J, Liao J, Yang Z, Wang S, Zhao Q. BiLSTM with multi-polarity orthogonal attention for implicit sentiment analysis. *Neurocomputing*. 2020 Mar 28;383:165-73.
- [2] Sahu TP, Khandekar S. A Machine Learning-Based Lexicon Approach for Sentiment Analysis. *International Journal of Technology and Human Interaction (IJTHI)*. 2020 Apr 1;16(2):8-22.
- [3] Nuruzzaman M, Hussain OK. IntelliBot: A Dialogue-based chatbot for the insurance industry. *Knowledge-Based Systems*. 2020 May 21;196:105810.
- [4] Han X, Zhou M, Turner MJ, Yeh T. Designing Effective Interview Chatbots: Automatic Chatbot Profiling and Design Suggestion Generation for Chatbot Debugging. *Human Factors in Computing Systems 2021* May 6 (pp. 1-15).
- [5] Chuan CH, Morgan S. Creating and evaluating chatbots as eligibility assistants for clinical trials: an active deep learning approach towards user-centered classification. *ACM Transactions on Computing for Healthcare*. 2020 Dec 30;2(1):1-9.
- [6] Fast E, Chen B, Mendelsohn J, Bassen J, Bernstein MS. Iris: A conversational agent for complex tasks. *Human factors in computing systems 2018* Apr 21 (pp. 1-12).
- [7] Touimi YB, Hadioui A, El Faddouli N, Bennani S. Intelligent chatbot-LDA recommender system. *International Journal of Emerging Technologies in Learning (iJET)*. 2020 Oct 19;15(20):4-20.
- [8] Pillai R, Sivathanu B. Adoption of AI-based chatbots for hospitality and tourism. *International Journal of Contemporary Hospitality Management*. 2020 Sep 10.
- [9] Khanuja S, Dandapat S, Sitaram S, Choudhury M. A new dataset for natural language inference from code-mixed conversations. *arXiv preprint arXiv:2004.05051*. 2020 Apr 10.
- [10] Benlahbib A, Nfaoui EH. A hybrid approach for generating reputation based on opinions fusion and sentiment analysis. *Journal of Organizational Computing and Electronic Commerce*. 2020 Jan 2;30(1):9-27.
- [11] Sojasingarayar A. Seq2seq ai chatbot with attention mechanism. *arXiv preprint arXiv:2006.02767*. 2020 Jun 4.
- [12] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. 2018 Oct 11.

## DESIGN AND DEVELOPMENT OF AI-BOT USING SEQUENCE TO SEQUENCE LSTM

---

- [13] Chakravarthi BR, Priyadharshini R, Muralidaran V, Jose N, Suryawanshi S, Sherly E, McCrae JP. Dravidiancodemix: Sentiment analysis and offensive language identification dataset for Dravidian languages in code-mixed text. *Language Resources and Evaluation*. 2022 Feb 4:1-42.
- [14] Lopatovska I, Griffin AL, Gallagher K, Ballingall C, Rock C, Velazquez M. User recommendations for intelligent personal assistants. *Journal of Librarianship and Information Science*. 2020 Jun;52(2):577-91.
- [15] Kunchukuttan A, Kakwani D, Golla S, Bhattacharyya A, Khapra MM, Kumar P. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages. *arXiv preprint arXiv:2005.00085*. 2020 Apr 30.
- [16] Yang Z, Qi P, Zhang S, Bengio Y, Cohen WW, Salakhutdinov R, Manning CD. HotpotQA: A dataset for diverse, explainable multi-hop question answering. *arXiv preprint arXiv:1809.09600*. 2018 Sep 25.
- [17] Moghe N, Arora S, Banerjee S, Khapra MM. Towards exploiting background knowledge for building conversation systems. *arXiv preprint arXiv:1809.08205*. 2018 Sep 21.

## LIST OF PUBLICATIONS

Santhosh A, Saranya T, Sundar S, Natarajan S. Deep Learning Techniques for Brain Tumor Diagnosis: A Review. In 2021 Fourth International Conference on Microelectronics, Signals Systems (ICMSS) 2021 Nov 18 (pp. 1-6). IEEE.