# Conversational Chatbot Using Deep Learning Neural Networks

Raksha Rawat[1], Rashi Dang[2]
Department of Computer Science, The NorthCap University, Gurgaon, Haryana, India
raksha18csu166@ncuindia.edu[1], rashi18csu167@ncuindia.edu[2]

---

**Abstract:**
A chatbot is a computer software that is used for interaction between a computer and a human being. Humans use different languages like English, Hindi, Spanish, etc. to communicate with one another, similar to that chatbots use natural language to communicate with humans. They are most widely used by organizations for their websites to assist users/customers/visitors with inquiries, information acquisition, and customer services. Other popular applications of chatbots include conversational agents such as Siri from Apple, Cortana from Microsoft, Google Assistant from Google, and Alexa from Amazon which are able to communicate with users as perform operations, functions and assist them with their daily tasks. Example, if we speak 'what are neural networks?' on Google Assistant, it will open up a google page for me with a search tab as 'what are neural networks?' and repeat the which it is getting. In this research paper, we perform research on the implementation and design of chatbots. We build a conversational chatbot that resembles a human being capable of interacting with humans in such a way that the user think they are actually talking to a human. This project uses deep learning neural networks as its foundation and creates a generative-based chatbot on Cornell Movie--Dialogs Corpus using Seq2Seq neural networks. During modeling, with the increasing number of epochs, the loss function decreases and the accuracy of the model increases. The loss function is 1.29 and the accuracy is 70% at the 40$^{th}$ epoch (last epoch).

---

## Introduction

A chatbot is a computer program that conducts a conversation via auditory or textual methods. These programs are created to simulate how a human would behave as a conversational partner. Chatbots are mainly used in dialog systems for various practical purposes including customer services and information acquisition. They are also used a lot in customer interaction, marketing on social network sites, and instant messaging to the clients. Chatbots can be used in many fields like medical fields for providing quick treatment or medications to people in emergency or people who are living in rural areas with no proper facilities of nearby clinics or hospitals. It can be used in websites for making it interactive to the users and answering their queries, in banking systems for providing better facilities, in customer care services to solve their issues, and many more areas. Providing people with 24/7 hrs of service. As technology is advancing, people demand more advanced facilities. A chatbot not only saves time for the customer/ user who has some queries but also makes it easier for the organization to work in a more efficient way.

There are two types of chatbot models:

1. **Retrieval based Chatbots:**
   Retrieval-based chatbots use a set of predefined input patterns and responses. It then applies some algorithms in order to select the appropriate response to the input question. They are used in making goal-oriented chatbots where there are a limited number of inputs and responses that can be generated, for example, to answer FAQ section questions. We can also customize the flow and tone of the chatbot in order to provide our customers with the best possible experience.

   Example,

```
{"intents": [
        {"tag": "greeting",
         "patterns": ["Hi there", "How are you", "Is anyone there?","Hey","Hola", "Hello", "Good day"],
         "responses": ["Hello, thanks for asking", "Good to see you again", "Hi there, how can I help?"],
         "context": [""]
        },
        {"tag": "goodbye",
         "patterns": ["Bye", "See you later", "Goodbye", "Nice chatting to you, bye", "Till next time"],
         "responses": ["See you!", "Have a nice day", "Bye! Come back again soon."],
         "context": [""]
        },
        {"tag": "thanks",
         "patterns": ["Thanks", "Thank you", "That's helpful", "Awesome, thanks", "Thanks for helping me"],
         "responses": ["Happy to help!", "Any time!", "My pleasure"],
         "context": [""]
        },
        {"tag": "noanswer",
         "patterns": [],
         "responses": ["Sorry, can't understand you", "Please give me more info", "Not sure I understand"],
         "context": [""]
        },
```

Figure 1: Dataset for Retrieval Based Chatbot

   This is the input dataset of a retrieval-based chatbot. Here, let’s suppose the user provides the input “Hi” or “Hello”, or something with the same semantic meaning, then our algorithm will identify the pattern, classify the input in the tag of “greeting” and then select the appropriate responses from the set of responses and provide the response as output. Similarly, it predicts the response for other inputs.

2. **Generative based Chatbots:**
   Generative models are different from retrieval-based models in a way that they do not depend on predefined responses. They generate new responses from scratch using the sequence-to-sequence model.

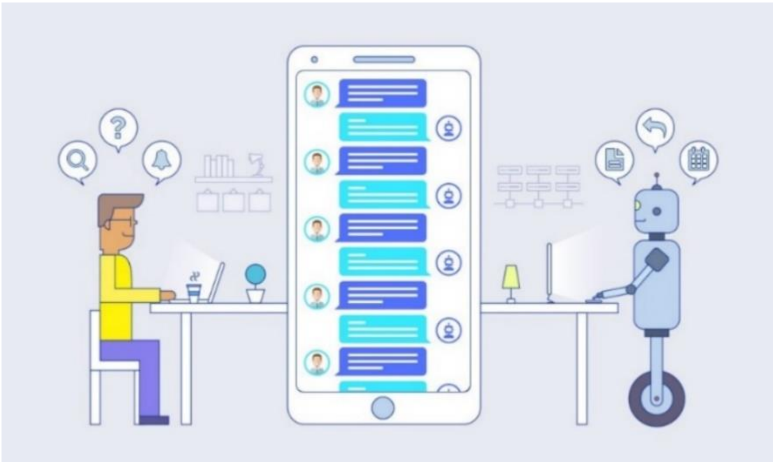

Figure 2: Chatbot

---

## Literature Review:

## Survey:

| S.No. | Author | Data | Method/Approach | Challenges | Result |
|---|---|---|---|---|---|
| 1. | Menal Dahiya (Maharaja Surajmal Institute) | Dataset: None<br><br>Preprocessing:<br>1) 2D string arrays applied to build database.<br>2) Rows in array used for request & response. | 1) Used simple pattern matching Algorithm<br>2) Modules used:<br>a) Chatbot()<br>b) Random()<br>c) AddText()<br>d) InArray() | 1) Inability to perform compound activities<br>2) Inability to answer to complex questions | Chatbot is:<br>1) Very simple<br>2) User friendly<br>3) Not very complicated |

| | | | | | |
|---|---|---|---|---|---|
| | | 3) Columns in array applied to save different types of questions | | | |
| 2. | Sonal Gupta, Rushin Shah, Mrinal Mohit, Anuj Kumar, Michael Lewis (Facebook AI Research) | Dataset: Self Acquired<br><br>Preprocessing:<br>1) Lowercasing<br>2) Removed punctuation marks and extra spaces<br>3) Tokenization | 1) Recurrent Neural Network Grammars (RNNG)<br>2) Seq2seq CNN<br>3) Seq2seq transformer<br>4) Seq2seq LSTM | | RNNG performs better than Seq2Seq models with an accuracy of 75% |
| 3. | Sasha Fathima Suhel, Vinod Kumar Shukla, Ved Prakash Mishra (Dubai, UAE) And Sonali Vyas (Dehradun, India) | Dataset: FAQ dataset<br><br>Pre-processing:<br>1) Lowercasing<br>2) Removed punctuation marks and extra spaces<br>3) Tokenization | A.L.I.C.E (Artificial Linguistic Internet Computer Enterprise)<br>a) Atomic categories<br>b) Default categories<br>c) Recursive categories | | 1) Some misunderstandings occur related to voice or text-based conversation.<br>2) Needs more information and further improvement to reduce the faults. |
| 4. | Raj Nath Patel, Rohit Gupta, Prakash B. | Dataset: Examples were used<br><br>Preprocessing: | Reordering approach<br>a. Noun Phrase Rules | | Addition of more reordering rules improves the translation quality. |

| | | | | | |
|---|---|---|---|---|---|
| | Pimpale and Sasikumar M CDAC Mumbai | 1) Tags created<br>2) Parsing | b. Verb Phrase Rules<br>c. Adjective and Adverb Phrase Rules<br>d. Preposition Phrase Rules | | |
| 5. | Vibhor Sharma, Monika Goyal, Drishti Malik | Dataset: None | 1) ELIZA (Rule Based)<br>2) ALICE (Artificial Linguistic Internet Computer Entity) | Why Alice is better?<br>1) Simple pattern matching algorithm<br>2) Recursive technique<br>3) Combines two answers if splitting happened within normalisation.<br>4) Easy and depend on depth first search. | 1) ALICE could not pass Turing test.<br>2) It is easier to make bots using ALICE than ELIZA as it is based on pattern matching approach. |
| 6. | Chaitrali S. Kulkarni, Amruta U. Bhavsar, Savita R. Pingale, Prof. Satish S. Kumbhar | Dataset: Referred different banks' websites and collected FAQs<br><br>Preprocessing:<br>1) Removed punctuation marks and extra spaces | Used NLTK library Classification techniques:<br>1) Decision Tree classifier<br>2) Bernoulli Naive Bayes Classifier<br>3) Gaussian Naive Bayes Classifier | | 1) Accuracy:<br>a) DT: 98.4%<br>b) BNB: 92.5%<br>c) GNB: 82.6%<br>d) KNN: 98.4%<br>e) MNB: 91.8%<br>f) RFC: 98.45%SVM: 98.45% |

| | | | | | |
|---|---|---|---|---|---|
| | | 2) Tokenization<br>3) Lemmatization<br>4) Vectorization | 4) K-nearest neighbor classifier<br>5) Multinomial Naive Bayes classifier<br>6) Random Forest classifier<br>7) Support vector machine | | 2) Bestalgorithms are Random Forest classifier and Support Vector Machine classifier. |
| 7. | Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, Wei Chu | Dataset: Chatlog of Alibaba's online customer service center<br><br>Preprocessing:<br>1) Flattened consecutive questions. | 1) A Seq2Seq based Rank Approac<br>2) QA Knowledge Base<br>3) IR Model<br>4) Generation based Model | | Launched AliMe Chat as online service and combined it with AliMe Assist |
| 8. | Dinesh Kalla (Colorado Technical University, USA) , Fnu Samaah (Northeastern Illinois University, USA) | Preprocessing:<br>1) Lowercasing<br>2) Tokenization<br>3) POS tagging | A mixed-method approach was used<br>1) Case I (partial answer exist):<br>a) Text extraction<br>b) Extracted Partial answer | | App provides user logins where their details will be stored in the chatbot database. Also, these details are saved for future reference. |

| | | | | | |
|---|---|---|---|---|---|
| | | | c) Cosine Similarity<br><br>2) Case II (partial answer doesn't exist):<br>a) Stemming | | |
| 9. | Amir Vakili, Azadeh Shakery | Dataset: Ubuntu corpus<br><br>Preprocessing: Used pre-processed dataset | Used BERT Bi-Encoder and BERT Bi-Encoder+CE model | | 1) Context Enrichment (CE) adds a relatively small amount of overhead and is faster than simple cross encoder architectures. |
| 10 | Wei-Nan Zhang, Ting Liu, Yifa Wang, Qingfu Zhu | Dataset: Collected from several Chinese online forums.<br><br>Preprocessing:<br>1) Converted to Lowercase<br>2) Removed punctuation marks and extra spaces<br>3) Tokenization | Personalized Response Generation:<br>1) Initialization and adaptation approach<br>2) RNN<br>3) B Encoder-decoder framework<br>Generation Quality Optimization:<br>1) Learning to Start (LTS) | | 1) Used imitation rate to analyze responses.<br>2) Evaluated PRM (Personalized Responding Model) with 5 volunteers |
| 11 | Anjana Tiha (The University | Dataset: Cornell Movie Subtitle Corpus | 1) Google's Neural Machine | 1) Limited performance during a | 1) Produced moderate results. |

| | | | | | |
|---|---|---|---|---|---|
| | of Memphis) | Preprocessing:<br>1) Remove Metadata<br>2) Remove Unsupported encoding format data<br>3) Separate data into 2 files, for dialogues and responses<br>4) Remove puctuation<br>5) Convert to lowercase<br>6) Remove consequent utterances<br>7) Tokenization | Translation (NMT) Model.<br>2) Seq2seq model. | long conversation.<br>2) Long training process<br>3) High power and processing demand. | 2) Some of the replies were repetitive and lacked proper relevancy.<br>3) Not suitable for imitating human interaction |
| 12 | Yogi Wisesa Chandra, Suyanto Suyantoa (School of Computing, Telkom University, Jl) | Dataset: Self Acquired from Telkom University Admissions<br><br>Preprocessing:<br>1) Lowercasing<br>2) Removing punctuation<br>3) Tokenization | Seq2Seq model using two-layered LSTM as encoder and decoder with and without attention mechanism | 1) Provide quick response to customers<br>2) Questions go unrequited | 1) BLEU score of 41.04 was produced by the model.<br>2) When attention mechanism technique was applied by reversing the sentences the BLEU score came up to 44.68. |
| 13 | Rui Xia, Zixiang Ding | Dataset: Self Acquired (benchmark ECE corpus) | Individual Emotion and Cause Extraction | 1) In traditional ECE model, emotion | 1) Measures:<br>a) F1 Score: 61.28% |

| | | | | | |
|---|---|---|---|---|---|
| | | Preprocessing:<br>1) Documents having two or more emotions are split into several samples such that each contains only one emotion.<br>2) Merge documents with same content into one document, and label each emotion, cause pair in the document. | 1) Independent multi-task learning:<br>a) Hierarchical Bi-LSTM network<br>b) Attention Mechanism<br>2) Interactive multi-task learning:<br>a) Enhanced version of the independent multi-task learning model<br>b) Inter-EC<br>c) Inter-CE<br><br>Emotion-Cause Pairing and Filtering:<br>1) Cartesian product to obtain the set of all emotions and causes.<br>2) A Logistic regression model to detect for each candidate pair whether | must be annotated before cause extraction, which greatly limits its applications and ignores the fact that they are mutually indicative<br>2) In current ECPE model, two-step strategy may not be a perfect solution as mistakes made in first step will affect the results of the second step. | Precision: 67.21%<br>Recall: 57.05%<br>2) Pair filtering improves the model performance.<br>3) Model performs better than traditional methods for ECE tasks without the need for emotion annotations |

| | | | | | |
|---|---|---|---|---|---|
| | | | they have a causal relationship.<br>3) Remove pairs with no causal relationship | | |
| 14 | Shashi Pal Singh, Ajai Kumar, Hemant Darbari, Lenali Singh, Anshika Rastogi, Shikha Jain (AAI, Center for development of Advanced Computing, Pune, India) | Dataset: None<br><br>Preprocessing:<br>1) Sentence Segmentation<br>2) Tokenization, etc. | Language Model:<br>1) RAE<br>2) RNN (LSTM, GRU)<br>3) Recursive NN<br><br>Joint Translation Prediction:<br>1) FNN<br>2) RNN<br>3) CNN | 1) Lack of vocabulary, data sparseness, maintain history of vector values etc.<br>2) Problem of gradient descent when RNN is used<br>3) Need of high computational power may require multiple GPUs. | 1) RNN, RAE gives better result in text processing as compared to other neural networks.<br>2) Word alignment, reordering, and language modelling can be performed with the help of a well-trained deep neural network. |
| 15 | David Oniani, Yanshan Wang | Dataset: COVID-19 Open Research Dataset (CORD-19)<br><br>Preprocessing:<br>1) Extracted abstract and main body of articles from every | 1) GPT-2 Language Model: generates the answer to the question<br>2) Filtering using regex and string manipulation to prune | 1) Used smaller dataset due to hardware constraints and difficulty in fine tuning.<br>2) The question | Medical experts evaluated the results in which BERT achieved best performance. |

| | | | | | |
|---|---|---|---|---|---|
| | | JSON file, combined them together, and used them as a corpus.<br>2) Word embedding | Prunes the responses<br>3) Filtering using semantic similarity to preserve sentences that are most semantical to the question.<br>a) Cosine Similarity<br>b) Inner product<br>4) Embeddings:<br>a) Tf-idf<br>b) BERT<br>c) BioBERT<br>d) USE | pool only consisted of 12 questions. | |
| 16 | Jiwei Li1, Will Monroe, Dan Jurafsky, Alan Ritter, Michel Galley3, Jianfeng Gao3 (Microsoft Research, Redmond, WA, USA) | Dataset: Subset of 10 million messages from OpenSubtitles dataset | 1) Seq2Seq model<br>2) Mutual information score as reward.<br>3) Backpropagating mutual information score helped to generate sequences with higher rewards.<br>4) Regularly updated parameters using | 1) Evaluation is difficult.<br>2) Can explore a very small number of candidates and simulated turns as number of cases increases exponentially.<br>3) Manually defined reward | 1) Model has tendency to end a sentence with another question to take the further.<br>2) Length of dialogue: Proposed RL model achieves best evaluation score of 4.48 number of simulations<br>3) Diversity: |

| | | | | | |
|---|---|---|---|---|---|
| | | | stochastic gradient descent.<br>5) Optimization:<br>a) Initialized policy model with parameters from mutual information model.<br>b) Maximized expected future reward using likelihood ratio trick.<br>6) Curriculum Learning:<br>a) Begin by simulating dialogue for 2 turns, then gradually increase the number of turns. | function cannot cover the important aspects that defines an ideal conversation. | RL model generates more varied responses as compared to mutual information and Seq2Seq model.<br>4) Human Evaluation:<br>a) RL model is not optimized to predict next utterance, but rather to increase long-term reward.<br>b) RL model produces responses that are significantly easier to answer than does the mutual information system. |

| | | | | | |
|---|---|---|---|---|---|
| 17 | Xiujun Liy, Lihong Liy, Jianfeng Gaoy, Asli Celikyilmaz, Yun-Nung Chen | Dataset: Conversational data was collected from Amazon Mechanical Turk<br><br>Preprocessing: Labeled 280 dialogues | Used reinforcement learning<br>1) Single layer LSTM<br>2) Policy learning<br>3) Natural Language Generation (NLG)<br>4) Error Model Controlling | | 1) Reinforcement learning system outperforms rule-based agents.<br>2) Different slot error types have different impacts on the RL agents.<br>3) RL agents are more robust to certain types of slot-level errors. |
| 18 | Qiming Bao, Jiamou Liu, Lin Ni | Dataset:<br>1) Knowledge graph:Health Navigator New Zealand<br>2) QA pair dataset: eHealth Forum, Question Doctors, and WebMD<br>3) HBAM: trained on Quora duplicate questions dataset<br><br>Preprocessing:<br>1) Embedding | 1) Entity extraction with Aho-Corasick algorithm.<br>2) One BiLSTM and one attention layer in Siamese framework.<br>3) Metrics used: Manhattan Distance | 1) Developing mechanisms so that system can understand natural language like humans.<br>2) Extracting relevant information from domain-specific database | HBAM performs better than MaLSTM and BERT models in all the datasets with an accuracy of 81.2%, 81.3%, 80.9% and 81.2% |

| | | | | | |
|---|---|---|---|---|---|
| 19 | Bang Liu, Di Niu, Haojie Wei, Haolan Chen, Yancheng He | Dataset: Reading comprehension datasets from SQuAD<br><br>Preprocessing:<br>1) Parsing<br>2) Chunking<br>3) Tokenization<br>4) Stemming<br>5) POS Tagging<br>6) Named Entity Recognition | 1) Seq2Sequence Model with attention mechanism and copy mechanism<br>2) GPT2 language model | 1) Extraction of good quality question answer pairs from unstructured data.<br>2) Problem of input volume explosion due to random sampling as most of such combinations would lead to meaningless questions.<br>3) No prior training dataset available for this particular task. | 1) CS2S-VR-ACS model performs the best.<br>2) GPT2-ACS achieves better METEOR score, while CS2S-VR-ACS performs better over BLEU and ROUGE-L.<br>3) Model generates diverse and high quality questions. |

## Summary

1. **A Tool of Conversation: Chatbot**
   This paper discussed a simple Chatbot that is capable of communicating with the user. The user can easily ask their query in human language and receive the information regarding that. This paper talked about how to design and the steps to create a Chatbot. There are a variety of methods used to design a chatbot due to which the development of the chatbot design grows at an unpredictable rate. The author mentions the selection of OS, selection of software, creating a chat, creation of the chatbot, and Pattern matching Algorithm. For the fundamental design, they first created a dialog box and the database. The module

consisted of a lot of functions like Chatbot(), Random(), Text(), Trim(), AddText() and array() using which a simple chatbot was created.

2. **Semantic Parsing for Task Oriented Dialog using Hierarchical Representations**
   This paper discussed Task-Oriented Parsing (TOP). In this, the hierarchical representation helps to analyze the semantics of complex nested queries. This technique helps in answering more questions. In this they first converted the sentences to Lowercase, removed the punctuation marks and extra spaces, and then Tokenized the sentence into small tokens. The models used were Recurrent Neural Network Grammars (RNNG), Seq2seq CNN, Seq2seq transformer, Seq2seq LSTM. After applying the models and training them the results showed that RNNG performs better than Seq2Seq models with an accuracy of 75%.

3. **Conversation to Automation in Banking Through Chatbot Using Artificial Machine Intelligence Language**
   Here the user enters the input in the messaging platform then the Natural language processing is done and the results are sent to the Bot Logic where the information source is stored and the machine learning algorithm and actions are applied to receive the desired output. This paper talked in depth about the advantages and growth of the chatbot. It also mentioned the growth of the banking industries and global online banking and how the use of chatbots would help in such industries. They also mentioned the Artificial Intelligence Markup Language (AIML), its basic elements, categories, and ways to create it. And lastly, they concluded their paper with the Turing test and the problems faced by it.

4. **Reordering rules for English-Hindi SMT**
   This paper mainly discussed the steps or logic applied while translating a sentence from English to the Hindi language. They used Statistical Machine Translation (SMT) approach for reordering the sentences. The reordering approach was used to understand the syntax in the sentence and reorder it into the Hindi language. Penn tags were created and a rich set of rules were made for better machine translation like Noun Phrase Rule, Verb Phrase Rule, Preposition Phrase Rule, Adjective and Adverb Phrase Rules. The results were shown using the BLEU, NIST, mWER, and mPER scores.

5. **An Intelligent Behavior Shown by Chatbot System**
   A chatbot is computer software that is capable of communicating with the user. This paper talked about the already existing system ELIZA and ALICE (Artificial Linguistic Internet Computer Entity). ELIZA uses a rule-based algorithm whereas ALICE is based on a pattern-matching approach. They have explained these two systems and their architecture in detail. They talked about the results of the Turing test on both of them. ELIZA could easily pass the test whereas ALICA failed to pass the test. Lastly, they also mentioned that

how ALICE is a better system than ELIZA to work with. One of the reasons was the pattern matching approach.

6. **BANK CHATBOT – An Intelligent Assistant System Using NLP and Machine Learning**
   This paper discussed the chatbot systems for Banks. It helps in solving customer queries in no time giving them a great experience. Earlier they used to call and have to talk to the customer service people to register the queries or either go to the bank and stand in line to get their work done. Chatbot makes it easier and comfortable for the users to work on. The authors mentioned the Architecture of the chatbot and the feedback system. For the implementation, they first prepared the dataset. For pre-processing, the used NLTK libraries performed vectorization and applied different classification models. Then they developed the learning model, tested the models, and according to the results they choose the 2 best classification approaches with gave the maximum accuracy.

7. **AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine**
   This paper discussed Seq2Seq rank approach and its implementation. They used the QA Knowledge base, IR model, and Generation model. They also talked about the attentive Seq2 Seq Rerank model. After applying and comparing these models with public chatbots they found that their chatbots gave better results. Later they released their chatbot which is called the AliMe Chat for the online services.

8. **Chatbot for Medical Treatment using NLTK Lib**
   Chatbots can be used in many fields. In this paper, they talked about chatbots used in medical fields for providing quick treatment or medications to people in emergency or people who are living in rural areas with no proper facilities of nearby clinics or hospitals. As the chatbots give 24/7 hours of service it helps people to receive prescriptions at any time of the day. They used the mixed method approach to generate the output. There were 2 cases. Case 1 when the partial answer existed and Case 2 when the partial answer does not exist. Based on this the models were created and later the results were evaluated.

9. **Enriching Conversation Context in Retrieval-based Chatbots**
   This paper discussed Retrieval-based Chatbots. They mentioned BERT Bi- Encoder and BERT Bi- Encoder + CE models. They talked about the architecture, implementation, and representation of the models. They also compared the two models in detail. Finally, they concluded that Context Enrichment was better and faster than the simple Cross encoder architecture.

10. **Neural Personalized Response Generation as Domain Adaptation**

This mainly discusses General Quality Optimization and Personalized Response Generation. For General Quality Optimization they used the data from a Chinese online forum and for Personalized Response Generation they used the personal chats of 5 volunteers. Applied RNN and Encoder-decoder approach for personalized response and for creating general responses they applied Learning to start (LST) approach. As they combined these two different ways to respond to the queries. The chatbot was well trained to answer the queries in a general way as well as was capable of responding in a more personal way so that the conversation could be made more natural.

11. **Intelligent Chatbot using Deep Learning**

In this research paper, the author developed an intelligent conversational agent using GNMT (Google's Neural Machine Translation) model. The author also talked in detail about the Seq2Seq model and encoder-decoder approach. Bidirectional LSTM, Beam Search and Neural Attention Mechanism was further used for the implementation of the chatbot. For user interaction, GUI is implemented using the PyQT module. This paper even discussed about the RNN Architecture & Deep Reinforcement learning algorithm and how it can be used for making long conversation chatbots. Also, training of the model takes a lot of time which makes it difficult to test for the right set of parameters for performance optimization.

12. **Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Model**

In this the author developed a Seq2Seq model for a university chatbot which can be used by the students or parents for asking admission related queries. They mention about the ALICE bot which uses the AIML approach and the pattern recognition technique. This QA system is built using the Seq2Seq model with an attention mechanism approach. A conversation dataset is used from the Telkom University Admissions. They used BLEU score to evaluate the results

13. **Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts**

This paper mainly discusses about the Emotion cause extraction (ECE) which aims at extracting potential causes that lead to emotional expressions in the text. They also mention the ECPE (emotion cause pair extraction) and its implementation. For the main approach Individual emotion and cause extraction was performed followed by Emotion-cause pairing and filtering. The model got an F1 score of 61.28%, precision of 67.21%, and recall rate of 57.05%. The model performs better than traditional methods for ECE tasks without the need for emotion annotations by an F1 score of 27.1%.

14. **Machine translation using deep learning: An overview**

Machine translation is a method to convert the source sentence from one language to another language with the help of computerized systems without the need for human assistance. This research paper talks about Machine translation, Deep Learning approach and Deep Neural Networks (DNN). Deep Learning approach was used in Machine translation followed by DNN in translation process in which Word alignment, Rule selection and reordering, Language modelling and Joint Translation were performed. The main models used in Deep neural networks were FNN, RNN, CNN, RAE, Recurrent NN, and Recursive NN. For error computation, Reconstruction error and reordering error were considered and different methods were used for calculating them.

15. **A qualitative Evaluation of Language Models on Automatic Question-Answering for COVID-19**

    In this research paper they developed a chatbot which is used to answer the queries related to COVID-19. This chatbot can be used by people for getting information related to covid like what is covid, how it attacks our body, its precautions, safety measure taken to avoid it, vaccines for coronavirus and many medical and healthcare information. For the implementation of the chatbot they used a GPT-2 language model followed by filtering based using regex and string manipulation and filtering using semantic similarity to the question to obtain the final response. Further, the approaches used were TF-IDF, BERT, BioBERT, and USE.

16. **Deep Reinforcement Learning for Dialogue Generation**

    This paper uses a Seq2Seq model approach to build the chatbot. Reinforcement learning for open domain dialogue is used in which Action, state, policy and reward are applied. Supervised learning and mutual information techniques were also used in this model. The authors also discuss about how the dialogue simulation will happen between the two agents. For automatic evaluation BLEU score was calculated. They also considered length of dialogues, diversity and human evaluation. They calculated BLEU score for RL model, Seq2Seq model & mutual information model and did a Qualitative analysis regarding them.

17. **End-to-End Task-Completion Neural Dialogue Systems**

    This paper talks about the chatbot which helps the user to book movie tickets. They used Neural dialogue system which included language understanding (LU) and Dialogue management (DM) in which dialogue state tracking and policy learning were implemented. For User Stimulation they used Natural Language Generation (NLG) approach. Error model controller included Intent-Level Error which had the functions Random error(), Within-group error() & Between-group error() followed by Slot-Level Error which contained Random error(), Slot deletion(), Incorrect slot value() & Incorrect slot() functions. They also mentioned about the end-to-end reinforcement learning technique.

This paper contains a detailed analysis regarding the different errors mentioned above and their outcomes. They also discussed about human evaluation and DQN agent.

18. **HHH: An Online Medical Chatbot System based on Knowledge Graph and Hierarchical Bi-Directional Attention**
This paper discusses about an online medical chatbot that helps the users to prescription based on their medical condition at any time of the day. In this they mentioned NLU and NLP. They also talked about different chatbots, Knowledge base storage & retrieval and Siamese based semantic sentence similarity. They made the HHH system architecture for the medical chatbot. This paper also contains information about knowledge graph architecture and Hierarchical BiLSTM Attention Model (HBAM). Lastly, they concluded that HBAM performs better than MaLSTM and BERT models in all the 4 datasets with an accuracy of 81.2%, 81.3%, 80.9% and 81.2%.

19. **Asking Questions the Human Way: Scalable Question-Answer Generation from Text Corpus**
In the beginning, this paper mentioned the CCS concepts in which NLP, NLG and machine translation algorithms are listed. They talked in detail about the Answer-Clue-Style-aware Question Generation (ACS-QG) and its implementation. First, they obtained the training data for question generation then they applied ACS – aware question generation which included Seq2Seq model, encoder which used bidirectional GRU, NER tag & POS tag and decoder approach. Further they talked about sampling inputs for question generation and data filtering for quality control. For evaluating the ACS- aware question generation they used BLEU score, ROUGE-L score and METEOR score.

---

## Related Works:

1. **Recurrent Neural Networks:**
RNNs are special Deep Learning Neural Networks that have loops in them. It takes in a sequence of inputs rather than single input (as in sequential networks) and predicts based on the context, sequential data, history, or patterns in the data. One more difference between sequential and RNNs is that RNN takes a fixed length of the input vector and we have to define input size before feeding it into the model. Due to its ability to store sequential data, we can use RNN for many recognition and prediction tasks like stock market prediction, speech recognition, music generation, etc.

**Deep Neural Network Architectures:**

a) **Sequential Networks:**

RNN architecture varies greatly from Sequential/ANN architecture. Sequential model consists of input layer, hidden layers, and an output layer that are fully connected to each other. Here, we feed data to the input layer, then some computations happen in the hidden layer (based on parameters we provide, such as activation function, number of neurons, epochs, etc.) which gives the output in the output layer. The model uses weights and biases in order to optimize model.

Forward Pass: The model chooses and applies some weights and biases, then calculates the output based on some activation function.

Backward Pass: The model calculates the errors, backpropagates, and updates its weights and biases so that it can get a minimized error in the next forward pass.

The model will first use forward pass then backward pass, then again forward pass, and so on until it gets an optimized output.

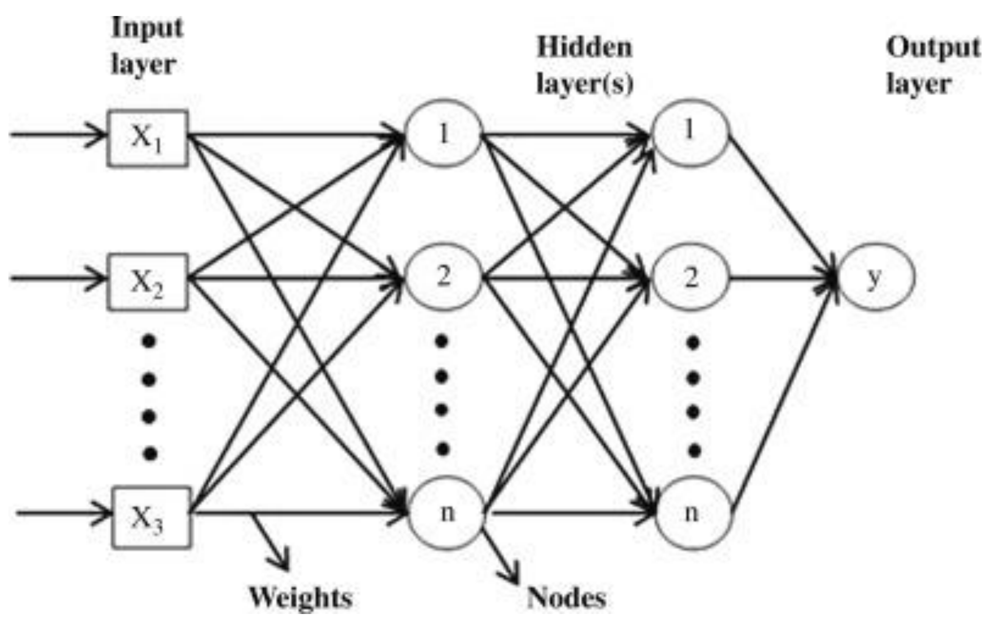

Figure 3: Sequential Network Architecture

**b) Recurrent Neural Network Architecture:**

RNNs have loops in them that allow for information to persist over time. We call these networks recurrent because the information passes from one time step to the next internally within the network.

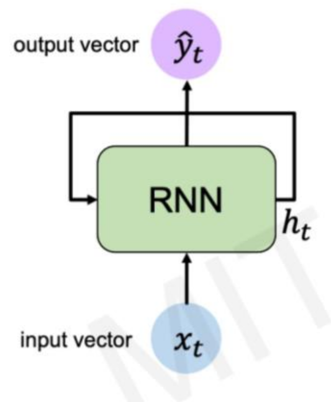


Figure 4: 1 to 1 RNN

At some time step (t), RNN takes an input $x_t$ and it computes the value $\hat{y}_t$ in the RNN layer which is then passed as the output of the network. Along with the computation of output, it also computes an internal state update $h_t$ and then passes this information about its internal state from the current time step of the network.

This cell state or internal state of the network is calculated by applying the following recurrence relation:
$h_t = f_W(x_t, h_{t-1})$
where $h_t$ represents current cell state, $f_w$ represents functions with weight W, $x_t$ represents input, and $h_{t-1}$ represents the previous state.

RNNs maintain this internal state $h_t$ and at each time step, it applies a function f along with some parameters and a weight matrix to update this state $h_t$. The key concept here is that this update is based on bot its previous state from the previous time step as well as the current input the network is receiving. The computation uses the same function $f_w$ and the same set of parameters at every time step. This state $h_t$ is updated with every time step as the sequence is processed further.

**Computational Graph of RNN Unrolled over time:**
RNNs have an input layer (x), hidden layers (h), and an output layer (y). The input layer takes an input in the form of a vector along with some weight (W) and bias. Then this input is passed into the hidden layer which consists of several RNN cells that calculate the output with the help of some activation function like a sigmoid, tangent, etc. along with its cell state. Then this output and current cell state is passed to the next hidden layer as the input, it then again calculates the output with the help of weight, bias, activation function, etc, along with its cell state and passes it to the next hidden layer. This process continues through all the hidden layers. The output layer receives the output calculated by the last hidden layer and applies some functions like softmax in order to generate the final output. The output vector from

the final output layer is then again fed into the input layer as an input vector. Hence, the sequence information is stored in the memory and utilized.

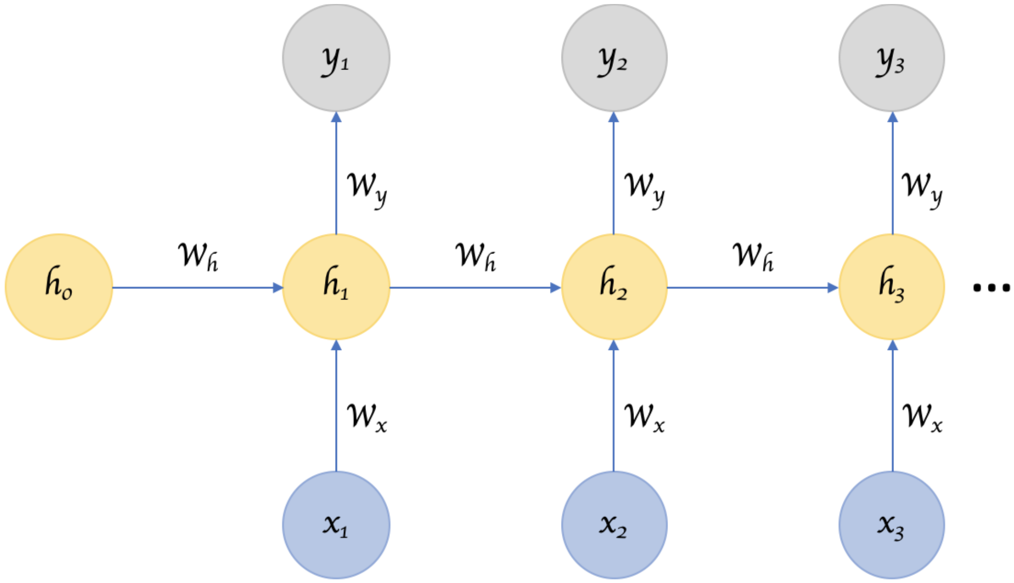


Figure 5: Computational Graph of Unrolled RNN

**Disadvantages:**

However standard RNNs have various disadvantages. RNNs are well suited for capturing short-term dependencies but they do not capture the long-term dependencies. For example, "I was born and brought up in Spain but last year I moved to Japan. I speak ________." Here in order to predict the next word, we need more context of the previous sentence. As this context gap increases, it becomes more difficult for standard RNNs to connect the dots and link relevant pieces of information together. Another problem faced by RNNs is the problem of vanishing gradients. As we repetitively use compute the gradients and use the activation function which takes the derivative of the gradient value, the gradient value becomes increasingly smaller and smaller moving towards zero until we can no longer train the neural networks. The problem of vanishing gradient can be solved by using the activation function Relu instead of tanh or sigmoid as it does not take any derivative but instead returns the input as output if the input is greater than 1 and returns 0 if the input is less than 0. The problem of vanishing gradient can also be solved by initializing the parameters, weight as an identity matrix, and bias as 0. This prevents the weights from shrinking to 0 during backpropagation or using special RNN cell LSTM. Also, standard RNNs stores the complete sequence information. This creates an information bottleneck in large datasets which can cause the network to perform poorly. All the above drawbacks of RNNs can be handled by using LSTM architecture.

2. **Seq2Seq Model:**
Sequence to Sequence models are based on RNNs and have an encoder-decoder architecture. Encoder RNN takes sentences as input and processes one word at a time at each timestep. It then converts this sequence of words into a fixed-size vector that contains only important information in the sequence. It helps in recognizing the context of the input given. The final hidden state called the context or thought vector shows the intention and summary of the input. This context then goes into the decoder as an input which then generates another sequence that represents the output.

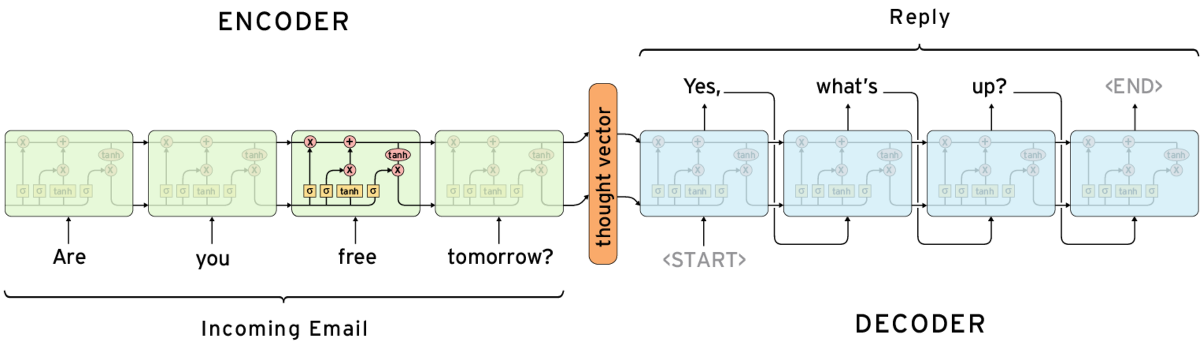


Figure 6: Encoder-Decoder Architecture

For example, when we pass the input sequence "Are you free tomorrow?" to the encoder-decoder network, the decoder generates words one by one at each time step of its iteration. After one iteration, the output sequence generated is "Yes, what's up?".

**Disadvantages:**
Though this model has a wide range of applications like machine translation, question answering, text summarization, and dialogue systems like chatbots, still there are a few disadvantages of using this model. The model cannot handle variable-length sequences while all sequence-to-sequence applications involve variable-length sequences. Also, the training process takes a long time as the decoder has to apply the activation function over a large vocabulary (given that we take a large dataset in order to get good results) for each word in the output. This will tremendously slow down the training process. Representation of words in the sequence is another problem. Using one-hot encoding creates sparse vectors due to large vocabulary and also no context or semantic meaning of the words is captured.

3. **Long-Short-Term-Memory (LSTM) Networks:**
Long-Short-Term-Memory networks are a special kind of RNNs that are capable of handling long-term dependencies. They are specifically designed to avoid the long-term dependency problem. For example, "I was born and brought up in Spain but last year I moved to Japan. I speak _______." In order to accurately predict the next word, more information is needed from the distant past which cannot be obtained from RNN. LSTMs

are also preferred over RNN as they solve the problem of vanishing gradient. As we repetitively use compute the gradients and use activation functions that take the derivative of the gradient value, the gradient value keeps on decreasing and starts to move towards zero and we can no longer train the neural networks. LSTMs selectively control the flow of gradients and information within its cell with the help of several gates thus solving the problem of vanishing gradients.

**LSTM Architecture:**

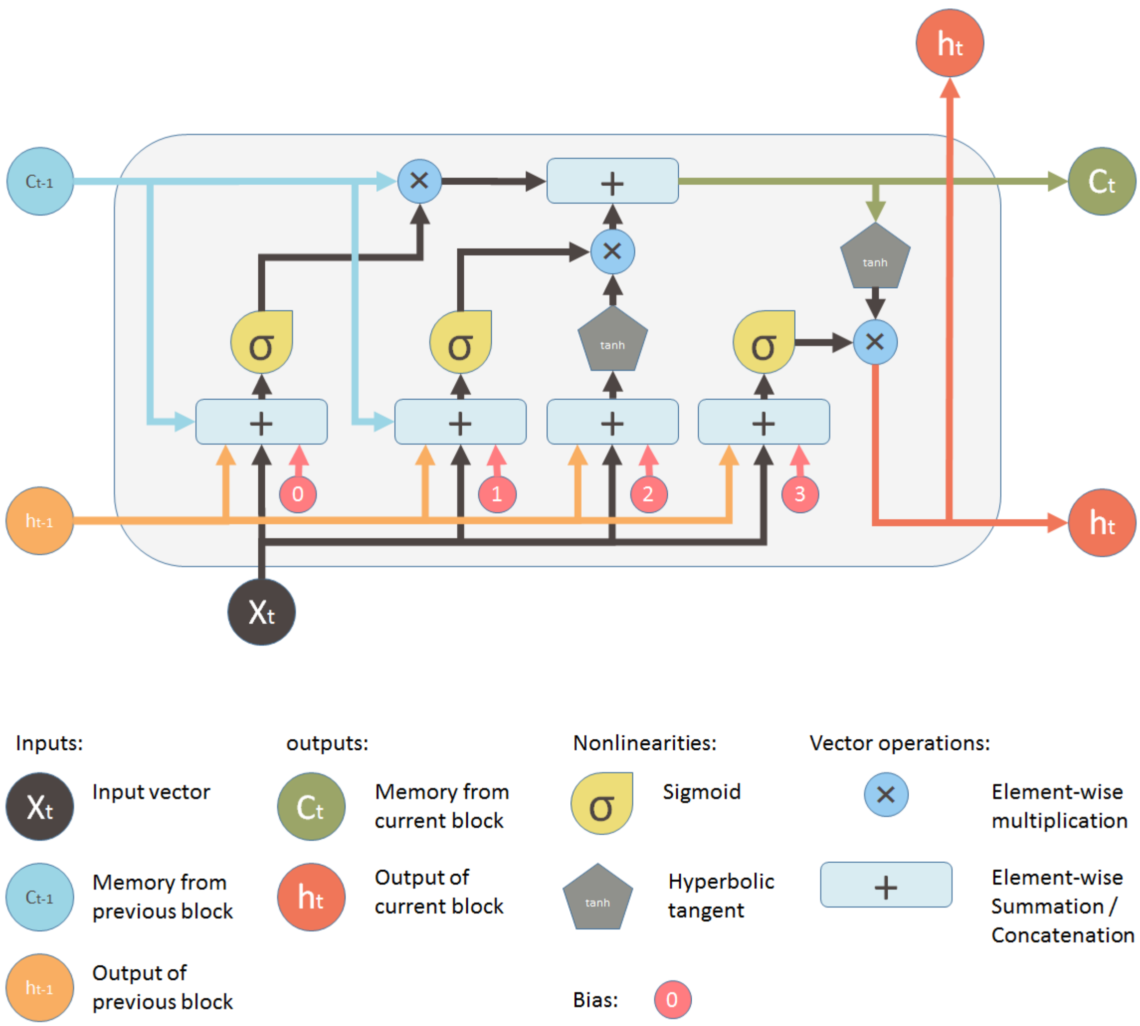


Figure 7: LSTM Architecture

LSTMs use gates to selectively control the flow of information. It contains forget gate, store gate, update gate, and output gate. Forget gate decides what information should be discarded and which should be kept from the cell state. It is the function of previous hidden state and current input. Its function is to forget the irrelevant history. The next gate is the store gate which decides what part of the new information is relevant and stores this information into its cell state. Next is the update gate which takes the relevant parts of the prior information and the current input and uses this to selectively update its state. At last there is output gate which gives the output and controls what information encoded in the cell state is sent as the next input in the next time step.

---

## Dataset

**Corpus: Cornell Movie-Dialogs Corpus**

**Link:** https://www.cs.cornell.edu/~cristian/Cornell_Movie-Dialogs_Corpus.html

**Description:**

The corpus contains a collection of fictional conversations from raw movie scripts along with their metadata. It has 220,579 conversational exchanges between 10,292 pairs of movie characters. There are a total of 9,035 characters from 617 different movies. Total utterances are 304,713. It also includes metadata of movies and characters.

All the files are in txt format and each field is separated by a '+++$+++':

**Corpus files:**

**Metadata:**

1) movie_titles_metadata.txt

   It has the metadata information of every movie title. It attributes are movieID, movie title, movie year, IMDB rating, and genres in format ['genre1', 'genre2', ……., 'genreN']

   Datafile:

```
movie_titles_metadata - Notepad
File  Edit  Format  View  Help
m0 +++$+++ 10 things i hate about you +++$+++ 1999 +++$+++ 6.90 +++$+++ 62847 +++$+++ ['comedy', 'romance']
m1 +++$+++ 1492: conquest of paradise +++$+++ 1992 +++$+++ 6.20 +++$+++ 10421 +++$+++ ['adventure', 'biography', 'dr
m2 +++$+++ 15 minutes +++$+++ 2001 +++$+++ 6.10 +++$+++ 25854 +++$+++ ['action', 'crime', 'drama', 'thriller']
m3 +++$+++ 2001: a space odyssey +++$+++ 1968 +++$+++ 8.40 +++$+++ 163227 +++$+++ ['adventure', 'mystery', 'sci-fi']
m4 +++$+++ 48 hrs. +++$+++ 1982 +++$+++ 6.90 +++$+++ 22289 +++$+++ ['action', 'comedy', 'crime', 'drama', 'thriller'
m5 +++$+++ the fifth element +++$+++ 1997 +++$+++ 7.50 +++$+++ 133756 +++$+++ ['action', 'adventure', 'romance', 'sc
m6 +++$+++ 8mm +++$+++ 1999 +++$+++ 6.30 +++$+++ 48212 +++$+++ ['crime', 'mystery', 'thriller']
m7 +++$+++ a nightmare on elm street 4: the dream master +++$+++ 1988 +++$+++ 5.20 +++$+++ 13590 +++$+++ ['fantasy',
m8 +++$+++ a nightmare on elm street: the dream child +++$+++ 1989 +++$+++ 4.70 +++$+++ 11092 +++$+++ ['fantasy', 'h
m9 +++$+++ the atomic submarine +++$+++ 1959 +++$+++ 4.90 +++$+++ 513 +++$+++ ['sci-fi', 'thriller']
m10 +++$+++ affliction +++$+++ 1997 +++$+++ 6.90 +++$+++ 7252 +++$+++ ['drama', 'mystery', 'thriller']
m11 +++$+++ air force one +++$+++ 1997 +++$+++ 6.30 +++$+++ 61978 +++$+++ ['action', 'drama', 'thriller']
m12 +++$+++ airplane ii: the sequel +++$+++ 1982 +++$+++ 5.80 +++$+++ 15210 +++$+++ ['comedy', 'romance', 'sci-fi']
m13 +++$+++ airplane! +++$+++ 1980 +++$+++ 7.80 +++$+++ 57692 +++$+++ ['comedy', 'romance']
m14 +++$+++ alien nation +++$+++ 1988 +++$+++ 6.10 +++$+++ 5590 +++$+++ ['crime', 'drama', 'sci-fi', 'thriller']
m15 +++$+++ aliens +++$+++ 1986 +++$+++ 8.50 +++$+++ 173518 +++$+++ ['action', 'sci-fi', 'thriller']
m16 +++$+++ amadeus +++$+++ 1984 +++$+++ 8.40 +++$+++ 99138 +++$+++ ['biography', 'drama', 'music']
m17 +++$+++ an american werewolf in london +++$+++ 1981 +++$+++ 7.50 +++$+++ 24443 +++$+++ ['horror', 'romance']
m18 +++$+++ american madness +++$+++ 1932 +++$+++ 7.40 +++$+++ 702 +++$+++ ['drama']
m19 +++$+++ american outlaws +++$+++ 2001 +++$+++ 5.70 +++$+++ 7425 +++$+++ ['action', 'western']
```

Figure 8: movie_titles_metadata.txt

2) movie_characters_metadata.txt

   It has the metadata information of every movie character. Its attributes are characterID, character name, movieID, movie title, gender, and position in credits.

Data file:

```
movie_characters_metadata - Notepad
File Edit Format View Help
u0 +++$+++ BIANCA +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ f +++$+++ 4
u1 +++$+++ BRUCE +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ ? +++$+++ ?
u2 +++$+++ CAMERON +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ m +++$+++ 3
u3 +++$+++ CHASTITY +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ ? +++$+++ ?
u4 +++$+++ JOEY +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ m +++$+++ 6
u5 +++$+++ KAT +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ f +++$+++ 2
u6 +++$+++ MANDELLA +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ f +++$+++ 7
u7 +++$+++ MICHAEL +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ m +++$+++ 5
u8 +++$+++ MISS PERKY +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ ? +++$+++ ?
u9 +++$+++ PATRICK +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ m +++$+++ 1
u10 +++$+++ SHARON +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ ? +++$+++ ?
u11 +++$+++ WALTER +++$+++ m0 +++$+++ 10 things i hate about you +++$+++ m +++$+++ 9
u12 +++$+++ ALONSO +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u13 +++$+++ AROJAZ +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u14 +++$+++ BEATRIX +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u15 +++$+++ BOBADILLA +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u16 +++$+++ COLUMBUS +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ m +++$+++ 1
u17 +++$+++ FERNANDO +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u18 +++$+++ ISABEL +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u19 +++$+++ MARCHENA +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ m +++$+++ 6
u20 +++$+++ MENDEZ +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u21 +++$+++ MOXICA +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u22 +++$+++ PINZON +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u23 +++$+++ SAILOR +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u24 +++$+++ SANCHEZ +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
u25 +++$+++ UTAPAN +++$+++ m1 +++$+++ 1492: conquest of paradise +++$+++ ? +++$+++ ?
```

Figure 9: movie_characters_metadata.txt

**Datasets:**

3) movie_lines.txt

It contains the actual text of each utterance. Its attributes are, lineID, characterID, movieID, character name, text of the utterance.

Datafile:

```
movie_lines - Notepad
File Edit Format View Help
L1045 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ They do not!
L1044 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ They do to!
L985 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ I hope so.
L984 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ She okay?
L925 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Let's go.
L924 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ Wow
L872 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Okay -- you're gonna need to learn how to lie.
L871 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ No
L870 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ I'm kidding.  You know how sometimes you just become this "persona"?  And yo
L869 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Like my fear of wearing pastels?
L868 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ The "real you".
L867 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ What good stuff?
L866 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ I figured you'd get to the good stuff eventually.
L865 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ Thank God!  If I had to hear one more story about your coiffure...
L864 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Me.  This endless ...blonde babble. I'm like, boring myself.
L863 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ What crap?
L862 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ do you listen to this crap?
L861 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ No...
L860 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Then Guillermo says, "If you go any lighter, you're gonna look like an extra
L699 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ You always been this selfish?
L698 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ But
L697 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ Then that's all you had to say.
L696 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Well, no...
L695 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ You never wanted to go out with 'me, did you?
L694 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ I was?
L693 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ I looked for you back at the party, but you always seemed to be "occupied".
L663 +++$+++ u0 +++$+++ m0 +++$+++ BIANCA +++$+++ Tons
L662 +++$+++ u2 +++$+++ m0 +++$+++ CAMERON +++$+++ Have fun tonight?
```

Figure 10: movie_lines.txt

4) movie_conversations.txt

It contains the structure of the conversations. Its attributes are characterID of the first character in the conversation, characterID of the second character in the conversation, movieID, list of the utterances that make the conversation in chronological order: ['lineID1', 'lineID2', ……, 'lineIDN']. The order has to be matched with movie_lines.txt to reconstruct the actual content.

Datafile:

movie_conversations - Notepad

File Edit Format View Help

```
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L194', 'L195', 'L196', 'L197']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L198', 'L199']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L200', 'L201', 'L202', 'L203']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L204', 'L205', 'L206']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L207', 'L208']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L271', 'L272', 'L273', 'L274', 'L275']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L276', 'L277']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L280', 'L281']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L363', 'L364']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L365', 'L366']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L367', 'L368']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L401', 'L402', 'L403']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L404', 'L405', 'L406', 'L407']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L575', 'L576']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L577', 'L578']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L662', 'L663']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L693', 'L694', 'L695']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L696', 'L697', 'L698', 'L699']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L860', 'L861']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L862', 'L863', 'L864', 'L865']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L866', 'L867', 'L868', 'L869']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L870', 'L871', 'L872']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L924', 'L925']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L984', 'L985']
u0 +++$+++ u2 +++$+++ m0 +++$+++ ['L1044', 'L1045']
u0 +++$+++ u3 +++$+++ m0 +++$+++ ['L49', 'L50', 'L51']
```

Figure 10: movie_conversations.txt

## Background information:

5) README.txt

It contains the information and details of all the files in the corpus.

File:

README - Notepad

File Edit Format View Help

```
Cornell Movie-Dialogs Corpus

Distributed together with:

"Chameleons in imagined conversations: A new approach to understanding coordination of linguistic style in dialogs"
Cristian Danescu-Niculescu-Mizil and Lillian Lee
Proceedings of the Workshop on Cognitive Modeling and Computational Linguistics, ACL 2011.

(this paper is included in this zip file)

NOTE: If you have results to report on these corpora, please send email to cristian@cs.cornell.edu or llee@cs.cornell.edu so we can add


Contents of this README:

        A) Brief description
        B) Files description
        C) Details on the collection procedure
        D) Contact


A) Brief description:

This corpus contains a metadata-rich collection of fictional conversations extracted from raw movie scripts:

- 220,579 conversational exchanges between 10,292 pairs of movie characters
- involves 9,035 characters from 617 movies
- in total 304,713 utterances
- movie metadata included:
        - genres
        - release year
        - IMDB rating
        - number of IMDB votes
        - IMDB rating
```

Figure 11: README.txt

6) raw_script_urls.txt
   It has the urls from which the sources were retrieved.
   File:

raw_script_urls - Notepad

File Edit Format View Help

```
m0 +++$+++ 10 things i hate about you +++$+++ http://www.dailyscript.com/scripts/10Things.html
m1 +++$+++ 1492: conquest of paradise +++$+++ http://www.hundland.org/scripts/1492-ConquestOfParadise.txt
m2 +++$+++ 15 minutes +++$+++ http://www.dailyscript.com/scripts/15minutes.html
m3 +++$+++ 2001: a space odyssey +++$+++ http://www.scifiscripts.com/scripts/2001.txt
m4 +++$+++ 48 hrs. +++$+++ http://www.awesomefilm.com/script/48hours.txt
m5 +++$+++ the fifth element +++$+++ http://www.scifiscripts.com/scripts/5thelement.txt
m6 +++$+++ 8mm +++$+++ http://www.dailyscript.com/scripts/eight-millimeter.html
m7 +++$+++ a nightmare on elm street 4: the dream master +++$+++ http://www.hundland.org/scripts/A-Nightmare-on-Elm-Street-4.txt
m8 +++$+++ a nightmare on elm street: the dream child +++$+++ http://www.hundland.org/scripts/A-Nightmare-on-Elm-Street-5.txt
m9 +++$+++ the atomic submarine +++$+++ http://leonscripts.tripod.com/scripts/ATOMICSUB.htm
m10 +++$+++ affliction +++$+++ http://www.dailyscript.com/scripts/Affliction.txt
m11 +++$+++ air force one +++$+++ http://www.dailyscript.com/scripts/AirForceOne_TXT.html
m12 +++$+++ airplane ii: the sequel +++$+++ http://www.dailyscript.com/scripts/Airplane2_script.htm
m13 +++$+++ airplane! +++$+++ http://www.dailyscript.com/scripts/Airplane_script.htm
m14 +++$+++ alien nation +++$+++ http://www.dailyscript.com/scripts/Alien_Nation_Bannon_Cameron_October_1987.html
m15 +++$+++ aliens +++$+++ http://www.horrorlair.com/scripts/aliens.html
m16 +++$+++ amadeus +++$+++ http://www.dailyscript.com/scripts/Amadeus.txt
m17 +++$+++ an american werewolf in london +++$+++ http://www.dailyscript.com/scripts/American%20Werewolf%20In%20London,%20An.txt
m18 +++$+++ american madness +++$+++ http://www.dailyscript.com/scripts/American_Madness.html
m19 +++$+++ american outlaws +++$+++ http://www.dailyscript.com/scripts/American_Outlaws.html
m20 +++$+++ american psycho +++$+++ http://www.dailyscript.com/scripts/american_psycho_unproduced.html
m21 +++$+++ antitrust +++$+++ http://www.dailyscript.com/scripts/Antitrust.txt
m22 +++$+++ austin powers: international man of mystery +++$+++ http://www.dailyscript.com/scripts/Austin%20Powers%20-%20Internat:
m23 +++$+++ the avengers +++$+++ http://www.dailyscript.com/scripts/Avengers.html
m24 +++$+++ bachelor party +++$+++ http://www.dailyscript.com/scripts/Bachelor%20Party.txt
m25 +++$+++ backdraft +++$+++ http://www.dailyscript.com/scripts/Backdraft%20(1991).txt
m26 +++$+++ bad lieutenant +++$+++ http://www.dailyscript.com/scripts/Bad%20Lieutenant.txt
m27 +++$+++ bamboozled +++$+++ http://www.awesomefilm.com/script/Bamboozled.txt
m28 +++$+++ barry lyndon +++$+++ http://www.dailyscript.com/scripts/BarryLyndon.html
m29 +++$+++ basic +++$+++ http://www.dailyscript.com/scripts/Basic.txt
```

Figure 12: raw_script_urls.txt

---

## Data Preprocessing

Cornell_Movie-Dialogs_Corpus dataset cannot be used as it is and needs to be cleaned and prepared to order to use for further computations.

**Preprocessing Steps:**

1) Creating a **nested list** of all conversations line ids.

2) Creating a **dictionary** to map each line id with its utterance.

3) **Create lists of questions and answers:**
   Seq2seq model requires inputs in form of questions and it generates outputs in the form of answers. Thus, we created a list of questions and answers. Our model will work better if we ask smaller length questions, as they'll be easy to process. So, we train the model on a fixed length of questions (maximum 13) and answers (maximum 11). To make the training faster, we take a subset of the corpus including 30000 questions and 30000 answers.

**4) Text Cleaning:**
   For this model, we have converted all the words to lower case and removed the punctuation marks using regex in order to clean the data.

Regex Rules:

```
# convert all words to lowercase and remove stop words using regex
def clean_text(txt):
    txt = txt.lower()
    txt = re.sub(r"i'm", "i am", txt)
    txt = re.sub(r"he's", "he is", txt)
    txt = re.sub(r"she's", "she is", txt)
    txt = re.sub(r"that's", "that is", txt)
    txt = re.sub(r"what's", "what is", txt)
    txt = re.sub(r"where's", "where is", txt)
    txt = re.sub(r"\'ll", " will", txt)
    txt = re.sub(r"\'ve", " have", txt)
    txt = re.sub(r"\'re", " are", txt)
    txt = re.sub(r"\'d", " would", txt)
    txt = re.sub(r"won't", "will not", txt)
    txt = re.sub(r"can't", "can not", txt)
    txt = re.sub(r"[^\w\s]", "", txt)
    return txt
```

Figure 13: Regex Rules

5) **Vocabulary Building:**

Deep learning algorithms work on integer data rather than string format so we need to convert them to integer values. So, for this purpose, we create a vocabulary. A vocabulary is a dictionary that has each and every word which is available in clean answer and clean questions with a unique value assigned to that particular word, count occurrence in this case. Format, {word(key): frequency count(value)}

6) **Remove less frequent words:**

There were many common English words like I, am, the, a, etc. which occur a lot in the data but not provide any valuable insight, thus it is important to remove them for effective preprocessing. In this, if the word count is less than 5 we remove that and if it is greater than 5 it will keep that.

7) **Tagging:**

Suppose our answer is hi, we will not just directly feed in high to our decoder model. We need to feed in a unique token that signifies the Start Of String (SOS) and a unique token that signifies the End Of String (EOS). <SOS > hi <EOS>

Different Tags:

<SOS>: Start of String

<EOS>: End of String

<PAD>: Padding

<OUT>: Output.

8) **Encoder and decoder inputs:**
Converting the list of questions and answers into integer values to create Encoder Inputs and decoder inputs.

---

# Approach:

## 1. Model Summary:

```
Epoch 40/40
938/938 [==============================] - 251s 267ms/step - loss: 1.2928 - acc: 0.7053
Model: "model"
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_2 (InputLayer)            [(None, 13)]         0
__________________________________________________________________________________________________
input_1 (InputLayer)            [(None, 13)]         0
__________________________________________________________________________________________________
embedding (Embedding)           (None, 13, 50)       151400      input_1[0][0]
                                                                 input_2[0][0]
__________________________________________________________________________________________________
lstm (LSTM)                     [(None, 13, 400), (N 721600      embedding[0][0]
__________________________________________________________________________________________________
lstm_1 (LSTM)                   [(None, 13, 400), (N 721600      embedding[1][0]
                                                                 lstm[0][1]
                                                                 lstm[0][2]
__________________________________________________________________________________________________
dense (Dense)                   (None, 13, 3027)     1213827     lstm_1[0][0]
==================================================================================================
Total params: 2,808,427
Trainable params: 2,808,427
Non-trainable params: 0
```

Figure 14: Model Summary

## 2. Methodology:

We'll have created a Seq2Seq model to implement our conversational chatbot.

1) **Input Layers:** We used two input layers one for encoder inputs and decoder inputs.

2) **Embedding Layer:** The embedding layer is used for dimensionality reduction. In our case, it compressed the vocabulary size 30000 to 50 values so it reduced the dimensionality. It takes three parameters, input dimension, output dimension, and input length. This embedding layer gives output in dimension [20, 13, 50] (50 being the output dimension which we have set). Next, we passed encoder and decoder placeholders to the embedding layer.

3) **Encoder LSTM Layer**: This is the encoder LSTM layer, it the encoder inputs as inputs. The number of cells used is 400 and it takes return state equals to true and returns sequence equals to true. We connected this LSTM layer with the encoder embedding in order to access the return states.

4) **Decoder LSTM Layer:** This is the decoder LSTM layer. It is similar to encoder LSTM cell but the difference here we will use encoder state as its initial state of decoder LSTM and in the vocabulary size we have to add one instead of reducing one because we are adding pad token extra in our vocabulary so we have to tell it by specifying plus one in the vocabulary size.

5) **Dense Layer:** It will output the probabilities. The number of probabilities will be equal to the vocabulary size. Since we need probability so we used the softmax activation function. Next, we connected this dense layer with the decoder output in order to get the final dense output.

6) **Model creation:** We used model class in order to create the model. It will take two arguments input data and output data. Input data includes encoder input and the decoder input while output data is the final dense output.

7) **Compile Model:** We compiled the model, setting the adam optimizer at categorical cross-entropy as a loss, and then fitted the model on our pre-processed data.

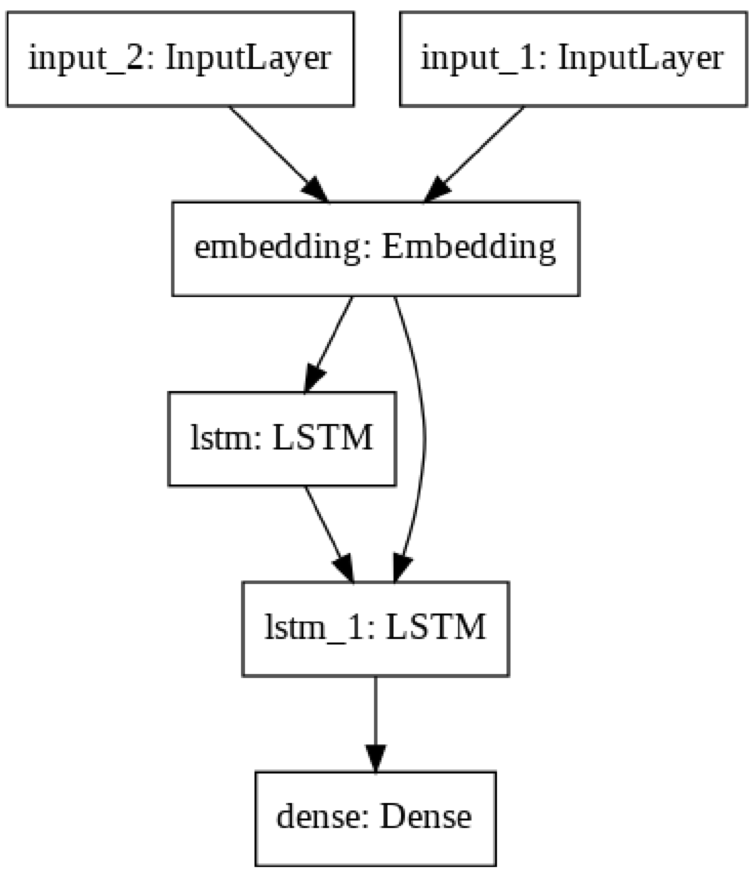


Figure 15: Model Layout

## Results:

1. **Experimental Results:**

The experimental results show that the accuracy keeps on increasing and the loss function keeps on decreasing with an increase with the number of epochs. We used a total of 40 epochs and it took 3 hours to execute the model. Epoch 1 had an accuracy of 46 % with a loss of 3.48, epoch 5 had an accuracy of 46 % with a loss of 3.48, epoch 10 had an accuracy of 56 % with a loss of 2.23, epoch 20 had an accuracy of 60% with a loss of 1.83, epoch 25 had an accuracy of 62 % with a loss of 1.67, and lastly in epoch 40 the accuracy increased to 70.05% and the loss function decreased to 1.29. Therefore, increasing the number of epochs resulted in higher accuracy and better communication.

The loss function used was categorical cross-entropy with the adam optimizer. The adam optimizer provides the best properties of the AdaGrad and RMSProp algorithms which helps to handle the sparse gradients on noisy problems. Finally, the accuracy of the model came out to be 70.05%. The higher the accuracy the better the results are which implies a more real conversation with the Chatbot.

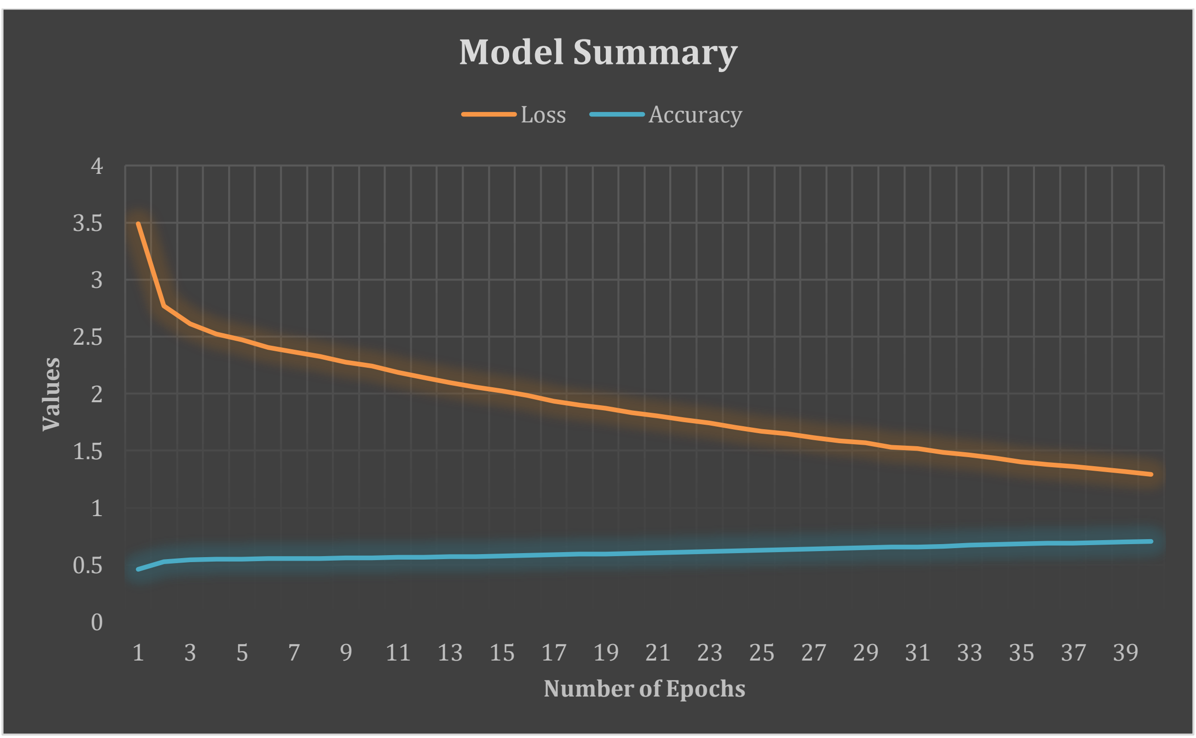


Figure 16: Loss and Accuracy Curve with respect to the number of epochs

## 2. Simulation Results:

The conversational chatbot was able to communicate with the user. The model training gave average results that need further improvement and more attention on the training parameters. Adding more quality data will further increase the performance of the model. This was an attempt to experiment with deep learning neural networks for the purpose of dialogue generation and management in order to develop an intelligent chatbot.

```
##########################################
#       start chatting ver. 1.0          #
##########################################
you : hi
WARNING:tensorflow:Model was constructed with shape (None, 13) f
chatbot attention :  hi
==============================================
you : hello
chatbot attention :  hello
==============================================
you : how are you
chatbot attention :  no
==============================================
you : what is your name
chatbot attention :  i dont know
==============================================
you : what
chatbot attention :  i dont know i dont know where to begin you
==============================================
you : where are you going
chatbot attention :  i dont know he is a <OUT>
==============================================
you :
```

Figure 17: Chatbot Demonstration Part 1

```
==============================================
you : what is your name
chatbot attention :  i dont know
==============================================
you : would you like to go watch a movie
chatbot attention :  i am sorry
==============================================
you : am i disturbing you
chatbot attention :  i am sorry
==============================================
you : byr
chatbot attention :  <OUT>
==============================================
you : bye
chatbot attention :  what
==============================================
you : see you later
chatbot attention :  i am sorry
==============================================
you : bye bye
chatbot attention :  what
==============================================
you : hello
chatbot attention :  hello
==============================================
you : q
chatbot attention :  <OUT>
```

Figure 18: Chatbot Demonstration Part 2

```
you : how are you
chatbot attention :  i am not a <OUT> and panties
==============================================
you : i want some food
chatbot attention :  i am sorry
==============================================
you : why
chatbot attention :  i dont know
==============================================
you : how can you help me
chatbot attention :  i dont know
==============================================
you : star wars
chatbot attention :  i am sorry
==============================================
you : what are you doing
chatbot attention :  i am not sure
==============================================
you : what is your hobby
chatbot attention :  i dont know
==============================================
you : who am i
chatbot attention :  <OUT> <OUT> <OUT>
==============================================
you : what is your name
chatbot attention :  i dont know
==============================================
you : would you like to go watch a movie
Executing (5m 44s) Cell > raw_input() > _input_request() > recv() > recv_multipar
```

Figure 19: Chatbot Demonstration Part 3

---

## Challenges and Limitations:

The main challenge in developing chatbot or dialogue generator lies in developing coherent dialogue generation system.

1. **Limited performance during a long conversation:** The model cannot sustain long conversations as a model is primarily designed for machine translation which does not consider the history of earlier conversations so it is not as effective for dialogue generation.

2. **Large Data Requirement:** To produce good results, we'll have to train the model on a very large dataset with good quality real-world conversations and data. Here we used a smaller dataset due to computation limitations and we can see that the chatbot did not give accurate responses in some cases.

3. **High Computational Time:** Training takes a long time, several hours to execute. This makes it difficult to experiment a lot with the hyperparameters.

4. **High Power and Processing Demand:** It demands multiple GPUs for its execution.

5. **Inaccurate Responses:** The model generated many general and repetitive responses because of a smaller and lack of good quality dataset.

## Conclusion

In this research paper, we experimented with Deep Learning Neural Networks in order to develop a conversational chatbot that can mimic human interactions. We created a conversational AI model which would answer the questions asked by the user. It is a simple bot with not many analytical skills but it is a good way to get started with NLP, neural networks and learn about chatbot architectures. Because of the smaller number of conversations in the dataset the model applied gives a finite number of answers which results in limited performance. Also, questions often go unrequited due to insufficient data.

## Future Work

The conversational chatbot can be further improved by providing high-quality real-life conversational datasets, which could mimic better human interaction. The training model should be trained with other hyper-parameters and different datasets for further experimentation. For example, the number of epochs can be increased and early stopping and dropout layers can be used in order to avoid overfitting and get the best possible results. In this way, we can hyperparameters can be further fine-tuned and get an optimized model. Different attention mechanisms can also be applied. Also, data preprocessing can be tried out using NLTK library. Multiple GPUs can be used for a faster training process. By implementing all these concepts, we will move towards an efficient chatbot system. The model can also be deployed on a voice-based chatbot to make it more interactive for the users.

## Acknowledgment:

We would like to express our sincerest regards to our faculty, Prof. Poonam Chaudhary for her valuable inputs, guidance, and constant support throughout the research.

## References:

[1] Menal Dahiya (Maharaja Surajmal Institute), "A Tool of Conversation: Chatbot", International Journal of Computer Sciences and Engineering, Review Paper Volume-5, Issue-5 E-ISSN: 2347-2693

[2] Sonal Gupta and Rushin Shah and Mrinal Mohit and Anuj Kumar (Facebook Conversational AI) Michael Lewis (Facebook AI Research), "Semantic Parsing for Task Oriented Dialog using Hierarchical Representations", arXiv:1810.07942v1 [cs.CL] 18 Oct 2018

[3] Sasha Fathima Suhel, Vinod Kumar Shukla, Ved Prakash Mishra (Dubai, UAE) And Sonali Vyas (Dehradun, India, "Conversation to Automation in Banking Through Chatbot Using Artificial Machine Intelligence Language", 978-1-7281-7016-9/20/$31.00 ©2020 IEEE

[4] Raj Nath Patel, Rohit Gupta, Prakash B. Pimpale and Sasikumar M CDAC Mumbai, “Reordering rules for English-Hindi SMT”, arXiv:1610.07420v1 [cs.CL] 24 Oct 2016

[5] Vibhor Sharma, Monika Goyal , Drishti Malik, “An Intelligent Behaviour Shown by Chatbot System”, International Journal of New Technology and Research (IJNTR) ISSN:2454-4116, Volume-3, Issue-4, April 2017 Pages 52-54

[6] Chaitrali S. Kulkarni, Amruta U. Bhavsar, Savita R. Pingale, Prof. Satish S. Kumbhar. (Department of Information Technology, College of Engineering Pune, India.), “BANK CHAT BOT – An Intelligent Assistant System Using NLP and Machine Learning”, International Research Journal of Engineering and Technology (IRJET)

[7] Minghui Qiu, Feng-Lin Li, Siyu Wang, Xing Gao, Yan Chen, Weipeng Zhao, Haiqing Chen, Jun Huang, Wei Chu Alibaba Group, Hangzhou, China, “AliMe Chat: A Sequence to Sequence and Rerank based Chatbot Engine”

[8] Dinesh Kalla , Vatsalya Samiuddin, “Chatbot for Medical Treatment using NLTK Lib”, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 22, Issue 1, Ser. III (Jan - Feb 2020), PP 50-56

[9] Amir Vakili and Azadeh Shakery University of Tehran, “Enriching Conversation Context in Retrieval-based Chatbots”, arXiv:1911.02290v1 [cs.CL] 6 Nov 2019

[10] Wei-Nan Zhang, Ting Liu, Yifa Wang, Qingfu Zhu, “Neural Personalized Response Generation as Domain Adaptation”, arXiv:1701.02073v2 [cs.CL] 2 Dec 2019

[11] Anjana Tiha (The University of Memphis), “Intelligent Chatbot using Deep Learning”, UID : U00619942 University of Memphis Spring, 2018

[12] Yogi Wisesa Chandra, Suyanto Suyantoa (School of Computing, Telkom University, Jl), “Indonesian Chatbot of University Admission Using a Question Answering System Based on Sequence-to-Sequence Mode”, 4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12-13 September 2019

[13] Rui Xia, Zixiang Ding (School of Computer Science and Engineering, Nanjing University of Science and Technology, China), “Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts”, Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1003–1012 Florence, Italy, July 28 - August 2, 2019.

[14] Shashi Pal Singh, Ajai Kumar, Hemant Darbari, Lenali Singh, Anshika Rastogi, Shikha Jain (AAI, Center for development of Advanced Computing, Pune, India), “MACHINE TRANSLATION USING DEEP LEARNING: AN OVERVIEW”

[15] David Oniani, Yanshan Wang (Mayo Clinic Division of Digital Health Sciences Rochester, MN, USA), “A qualitative Evaluation of Language Models on Automatic Question-Answering for COVID-19”, arXiv:2006.10964v2 [cs.IR] 23 Jun 2020

[16] Jiwei Li1, Will Monroe, Dan Jurafsky (Stanford University, Stanford, CA, USA), Alan Ritter (Ohio State University, OH, USA), “Deep Reinforcement Learning for Dialogue Generation”, arXiv:1606.01541v4 [cs.CL] 29 Sep 2016 Michel Galley3, Jianfeng Gao3 (Microsoft Research, Redmond, WA, USA), “”

[17] Xiujun Liy, Lihong Liy, Jianfeng Gaoy, Asli Celikyilmaz (Microsoft Research, Redmond, WA, USA) and Yun-Nung Chen (National Taiwan University, Taipei, Taiwan), "End-to-End Task-Completion Neural Dialogue Systems", arXiv:1703.01008v4 [cs.CL] 11 Feb 2018
[18] Qiming Bao, Jiamou Liu (The University of Auckland, Auckland), Lin Ni (The National Institute for Health Innovation (NIHI), Auckland), "HHH: An Online Medical Chatbot System based on Knowledge Graph and Hierarchical Bi-Directional Attention", arXiv:2002.03140v1 [cs.CL] 8 Feb 2020
[19] Bang Liu, Di Niu (University of Alberta, Edmonton, AB, Canada), Haojie Wei, Haolan Chen, Yancheng He (Platform and Content Group, Tencent, Shenzhen, China), "Asking Questions the Human Way: Scalable Question-Answer Generation from Text Corpus", arXiv:2002.00748v2 [cs.CL] 5 Mar 2020.