EAI Endorsed Transactions
on Scalable Information Systems

Research Article EAI.EU

Evaluating Performance of Conversational Bot Using Seq2Seq Model and Attention Mechanism

Karandeep Saluja[1], Shashwat Agrawal[2], Sanjeev Kumar[3*] and Tanupriya Choudhury[4]

[1,2,3] School of Computer Sciences, University of Petroleum and Energy Studies (UPES), Dehradun, Uttarakhand, 248007, India.
[4]Graphic Era Deemed to be University, Dehradun, 248002, Uttarakhand, India.

Abstract

The Chat-Bot utilizes Sequence-to-Sequence Model with the Attention Mechanism, in order to interpret and address user inputs effectively. The whole model consists of Data gathering, Data preprocessing, Seq2seq Model, Training and Tuning. Data preprocessing involves cleaning of any irrelevant data, before converting them into the numerical format. The Seq2Seq Model is comprised of two components: an Encoder and a Decoder. Both Encoder and Decoder along with the Attention Mechanism allow dialogue management, which empowers the Model to answer the user in the most accurate and relevant manner. The output generated by the Bot is in the Natural Language only. Once the building of the Seq2Seq Model is completed, training of the model takes place in which the model is fed with the preprocessed data. During training it tries to minimize the loss function between the predicted output and the ground truth output. Performance is computed using metrics such as perplexity, BLEU score, and ROUGE score on a held-out validation set. In order to meet non-functional requirements, our system needs to maintain a response time of under one second with an accuracy target exceeding 90%.

Keywords: Seq2Seq, Attention Mechanism, Perplexity, BLEU, ROUGE

Received on 10 December 2023, accepted on 11 March 2024, published on 18 March 2024

doi: 10.4108/eetsis.5457

*Corresponding author. Email: sanjeevkumar@outlook.in

1. Introduction

Artificial Intelligence is getting involved in our daily life with humans more and more, the line between humans and AI is getting bluer day-by-day. Today, AI can do everything that humans can from manufacturing to playing sports, from problem solving to writing code, one of the greatest example of this Human-Computer Interaction(HCI) [1]. It is a system which is able to understand the texts posed by humans in Natural Language, and it is able to answer them in the most accurate manner in the Natural Language itself [2]. In the Dictionarium, a chatbot is defined as "A computer Program which is able to understand the context of the user text and reply according to context only" [3].Chatbots can mimic humans, entertain humans, but their usage is not limited up to here only. They can be used in a wide number of applications like healthcare industry, education industry and so on. They became so popular as they have so many advantages and are easy to use for the users. They are platform independent and use APIs, they are instantly available to the users without any installation. Multiple conversations can be carried out in a single Chatbot, as most of the Chatbots are open-domain models. Chatbots use multiple technologies in order to make them powerful, technologies that are included are NLP, Information Retrial and ML.

2. Problem Identification

The main objective of Seq2seq Model with the Attention Mechanism is to build a model which is able to understand the context and tone of the user input and is able to reply to the user in the same context and tone accurately. The aimof the system is to develop a human-to-human-like conversation. The specific objectives of the system are as follows:

i. To develop a Seq2Seq Model with an Attention Mechanism capability that can effectively encode userinputs and generate appropriate responses.
ii. To pre-process and curate a large and diverse dataset of conversational exchanges to train and validate the Seq2Seq model.
iii. Fine-tuning the hyperparameters of Seq2Seq mode in order to optimize its performance and achieve highaccuracy.
iv. Integrate the response generation module with the dialogue management system to produce contextually appropriate responses derived from the conversation history.
v. Create a response postprocessing module that transforms the numerical output of the model into natural language text, ensuring coherence and grammatical correctness in the generated response.

In general, the Conversational Bot employing the Seq2Seq Model and Attention Mechanism strives to deliver a conversational experience of high quality, comparable to human-to-human interactions.

3. Literature Survey

Alan Turing introduced the concept of a chatbot with the Turing Test in 1950, posing the question "Can machines think?" [4]. The first recognized chatbot, Eliza, emerged in 1966 as a psychotherapist, responding to user input with questions using a basic pattern-matching and template-based response mechanism[7]. While Eliza's conversational abilities were limited, it provided a novel experience for users unaccustomed to interacting with computers, inspiring further chatbot development [5]. In 1972, PARRY, a chatbot with a personality, was developed as an improvement over Eliza [9]. By 1995, the chatbot ALICE had won the Loebner Prize in 2000, 2001, and 2004, becoming the first computer to be ranked as the "most human computer" [10]. ALICE employed a straightforward pattern-matching algorithm that enabled developers to define the chatbot's knowledge [10]. In 2001, chatbots like Smarter Child [12] were introduced and made accessible through messenger applications. The subsequent evolution of chatbots included the development of virtual personal assistants such as Apple Siri, Microsoft Cortana, Amazon Alexa, Google Assistant, and IBM Watson.

The revenue of the global chatbot market showcases noteworthy growth from 2022 to 2032, according to Fig 1 [18]. Predictions indicate that the market is poised to achieve substantial revenue of 454.8 million U.S. dollars by 2027, reflecting a considerable rise from the reported 40.9 million dollars in 2018. This upward trajectory underscores the expanding influence and economic significance of the chatbot industry on a global scale. Thus, it becomes necessary to have a deep understanding of the working of a chat-bot [18].

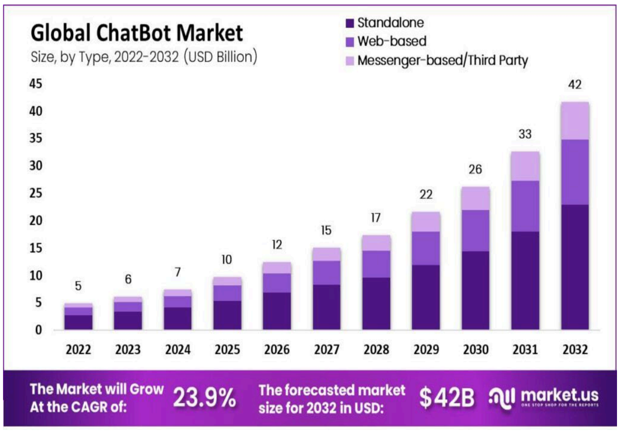

Figure 1: Revenue: Global Market of Chat Bot [18]

4. Associate with Chat-Bots

Chat Bots are easy to use, as the conversation between user and bot takes place in the Natural Language, it is coinvent for the user to perform the conversation, more over with the help of Bot many tasks can be performed faster and conveniently.

Chat Bot can be used for an open or closed domain, building a closed domain is easy, as our Bot will be having knowledge about a specific topic only, such Bots are useful for specific applications like for hospitals, education institutional, and so on. On the other hand Open-Domain have the knowledge of all the topics, and they are very powerful, they can be used for text generation, report generation, understanding of a topic, and so on [3].

There are many companies who want to provide customer service available for 24 * 7, for that company have to employ people and that's a costly solution, now the companies can provide these services with the help of Bots in lessexpensive way. From your home to your mobile, from your educational institutions to your factory, from a product toservice, everything can be handled by the Chat Bots now [3].

Using a Chat Bot is as customer service might seems an absurd solution, but in actual the Chat Bots are very powerful,with the help of NLP Techniques they are able to understand the sentiment of the text given by the user, with this theyare able to perform the conversation in the same context. Which makes them a good solution to for customer services [3]. Trust on Chat Bot depends upon how well they are able to do the conversation with the human i.e., the tone, formalnessof answers, and the context of the answers, in order to make more and more accurate, they require training, more they are trained on the diverse corpus, more they become powerful. Note It doesn't matter how much you train the Bot, they still lack the empathy that human beings have. With the advancement of technology, it will be possible to developthe empathy in them [3]. Moreover it is necessary to note while building a Chat Bot, we need to take care that our Bot should not be gender biased or racist in any way [3]. With the advancement of the technology more and more powerful Bots are been build, and soon there will be time, that Bots will be replacing the Human in every sector ofwork [3].

5. Existing System Issue

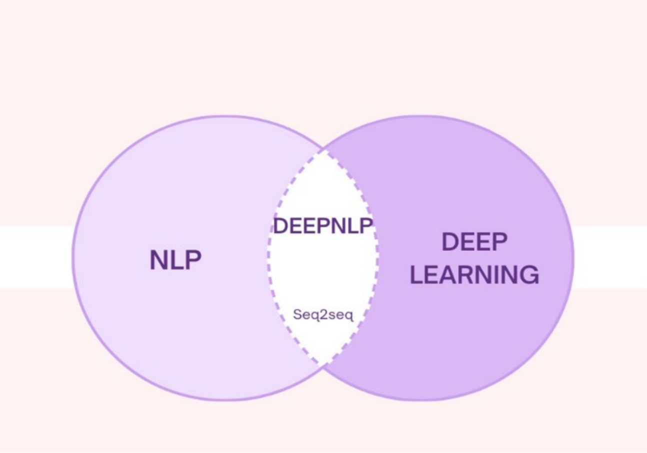

Figure 2. Inter-section of NLP and Deep Learning

Natural Language Processing (NLP) is a domain within artificial intelligence that focuses on comprehending and generating natural language, as humans speaks unstructured language, and machine speaks structured language only, to make machine understood the unstructured language spoken by human to machine, NLP is used.
Deep Learning is a subset of ML, their specialty is handling large amounts of data, and extracting the important features out of the dataset.
When we do natural processing with the help of deep leaning concepts then it becomes Deep Learning Natural Processing.

Classical vs Deep Learning Mode

1. If/Else Rules (Chat Bot)
   It is an example of Natural Language.
   This was the way that we used to create chat Bots, back in the day. They entail a huge list of possible questions and answers.
   So, somebody in the chat asks the question, so we need to record the answer to those questions in our model. Because of this, this type of Chat Bot has very limited limitations. Such a Mechanical Approach doesn't result in anything Human.
2. Audio Frequency Components Analysis (Speech Recognition): It's an example of Natural Language Processing.
   So, in essence, what happens is we look at the sound wave of somebody talking and then compare this frequency with the pre-recorded frequencies. So, we know certain combinations of frequencies mean this type of word. In such types of models, we are not doing any Neural Computations, we are not creating any Neural Networks.
   We are just doing Mathematical Calculations around the frequencies that we can observe comparing them to the Mathematical Calculations we have in our library of Pre-Analyzed Frequencies. Then we match them up and find the word.
3. Bag of Words Model (Classification)

Table 1: Classification results of Bag of Words Model

Comments	**Pass/Fail**
Great Job	1
Amazing Work	1
Well Done	1
Very Well Written	1
Poor Effort	0
Try Harder	0
Could have done Better	0

We have teachers who have checked some answer sheets and have left some comments there.
So, this model will try to associate the words with the grades, like how many times good comes with a pass (1) and how often great comes with a failure (0).
And this model is going to remember this associativity, and whenever similar words come, it is going to give an output.
Limitations:

- Fixed-size input.
- Doesn't take word order into account which leads to data loss.
- Fixed-size output.

6. Proposed System Design

Seq2Seq

Figure 3 illustrates a sample implementation of a standard Seq2Seq model, featuring word embeddings and an attention mechanism.

The key components in the model:

Encoder: The encoder is a stack of RNNs, it is responsible for understanding the text given by the user, it consists of many LSTM cells, and each cell is responsible for handling each token. The output of one cell becomes the input for the next cell.

Encoder Vector: The Encoder produces a fixed-length vector that characterizes the user text.

Decoder: The decoder is a stack of RNN, which is responsible for generating the text, from the features found by the Encoder, Encoder Vector is the first input for the first Decoder Cell, and then the output of each Decoder cell becomes the input for the next Decoder Cell.

Attention Mechanism: As the Encoder's output is a fixed-length vector, it becomes challenging for the decoder to assimilate all pertinent information from this vector. The remedy lies in an attention mechanism. This mechanism empowers the decoder to concentrate on crucial features during text generation. It achieves this by computing weights that indicate their significance for the ongoing decoding phase. By enabling the decoder to prioritize relevant segments, the model can produce more precise results.

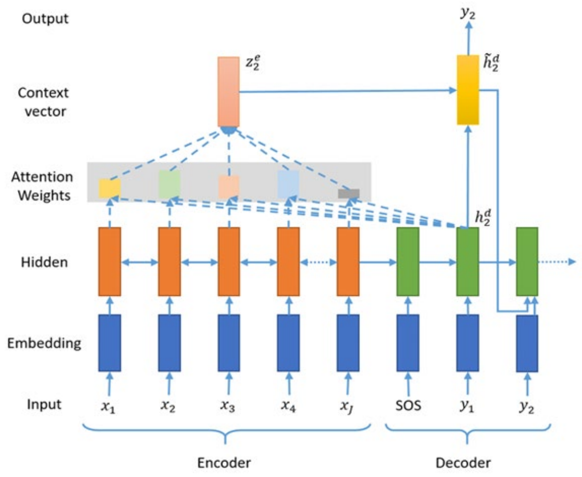

Figure 3: Design Architecture: An attention-based seq2seq model [16]

Initially the user input is given to the Encoder, where sentence tokenization is done, after tokenization, each token(word) is given to each encoder cell, since machine cannot understand the natural language spoken by the humans, so each tokens have to be converted into numerical format, process of converting the words into their numerical format is known as Vectorization, after Vectorization, each numerical-word is feed to each Encoder cell, where the output of each cell is the input for the next cell, in this way, Encoder is able to understand the context and sentiment of user input, last encoder cell output is the Encoder Vector which is the fixed length Vector describing about the features of User Input, this Encoder Vector becomes the input for the first Decoder Cell, each decoder cell is responsible for generating the text out these features given by the encoder, for decoder it's difficult to generate thetext from the fixed length Vector, here the Attention Mechanism helps the Decoder, to focus only on the relevant features which decoding/generating the text, the output of each decoder cell becomes the input for the next cell.
In this way the model is able to generate more accurate results.

7. Context Setting

First let's understand what Context Setting in the Chat-Bots is. Context Setting in the Natural Language Chat-Bots is the process of establishing and maintaining the ongoing understanding of the conversation, so that relevant and coherent responses can be provided to the users. The process involves keeping the track of the previously generated output, current user's input, so that the Chat-Bot can maintain the meaningful flow of the conversation. Context Setting is necessary for the Chat-Bots in order to generate the human-like conversations, without this the Chat-Bots will treat each user input has an isolation, leading to disconnected and irrelevant conversations. Context Setting can be implemented with the help of RNN, and Attention Mechanism in the Chat-Bot's Architecture. Such implementation allows the model to retain the sequential information and past interaction with the user.

Coming to the Seq2Seq with the Attention Mechanism Chat-Bot.
Let's understand all the parts of Seq2Seq with Attention Mechanism Chat-Bot again in the context of the Context Setting.

1. Encoder's Context Setting:
Upon receiving the user's input, the given Input Sequence is processed by the Encoder, generating a fixed-length context vector. This context vector is useful as it serves as a summary of the user's given input. The Encoder considers the entire given user's input sequence, while producing a context vector that encapsulates pertinent and significant details of the user's input. The Context Vector holds significance as it encapsulates crucial information from the user's input.

2. Attention Mechanism's Context Setting:
The Attention Mechanism improves the context-setting procedure by allowing the model to focus on specific and pertinent segments of the input sequence while generating the output sequence. Instead of forcing the decoder to rely solely on the fixed-length Context Vector, the Attention Mechanism guides the decoder to concentrate on various segments of the input sequence according to their relevance in generating the current output sequence. This method assists the model in preserving previously generated output and understanding the context of the user's input.

3. Decoder's Context Setting:
During the decoding process the decoder uses the produced Context Vector and produces one token at a

time. The Attention Mechanism, during the decoding process calculates the attention score for each input sequence token indicating how much a particular input token is important for the current output sequence token. By combing the Context Vector and Attention Mechanism the decoder becomes capable of maintaining and generating the relevant conversation.

4. Bidirectional Encoder:
By making the Encoder Bi-directional in nature the model becomes more capable of producing human-like conversation. This enables the model to capture dependencies from both past and future words in the user's input.

8. Context Setting Parameter

- **Architecture of the Encoder and Decoder:**

  The configuration of the encoder and decoder components in the "Seq2Seq model" is determined by this parameter. Common architectures involve "RNNs", "LSTMs", or "GRUs", each exhibiting different capabilities in capturing sequential dependencies within input data.

- **Dimensions of Hidden States:**

  The hidden state dimensions establish the size of the internal memory representation in both the encoder and decoder. Larger dimensions can capture more intricate patterns but may necessitate increased computational resources.

- **Number of Layers in Encoder and Decoder:**

  Defining the layer depth in the encoder and decoder has an impact on the model's ability to capture intricate patterns. A higher number of layers enables the model to learn more complex hierarchical representations.

- **Type of Attention Mechanism:**

  This parameter governs the type of attention mechanism employed in the "Seq2Seq model". Options include additive, multiplicative, or a hybrid, influencing how the model concentrates on specific segments of the input sequence during decoding.

- **Size of Attention Hidden Layer:**

  The attention hidden size determines the dimensionality of the attention vector, affecting the model's capability to assign distinct weights to different parts of the input sequence. This dimension is crucial for capturing nuanced relationships.

- **Attention Dropout:**

  To prevent overfitting during training, attention dropout is introduced. The specified dropout rate regulates the regularization applied to the attention mechanism, influencing the model's generalization to unseen data.

- **Functions for Attention Alignment:**

Alignment functions define how the model computes alignment scores between encoder and decoder hidden states during attention calculation. Common functions include dot product, additive, and scaled dot-product attention, each influencing how the model attends to relevant information in the input sequence.

9. Methodology

The process of developing a chatbot with a "Seq2Seq Model" and "Attention Mechanism" encompasses several key stages. Firstly, an extensive dataset of conversation pairs is collected, covering diverse topics and sentence structures. This gathered data then undergoes careful pre-processing, involving tasks such as tokenization and organizing into input-target pairs. The core of the methodology revolves around constructing the "Seq2Seq model" with an "Attention Mechanism", employing an "Encoder-Decoder Architecture" to comprehend input sequences and generate appropriate responses. Training the model, typically using datasets like the "Cornell Movie Corpus", involves splitting the pre-processed data into training and validation sets and fine-tuning hyperparameters. Post-training, the model undergoes rigorous testing on a separate dataset to evaluate its generalization capabilities. An iterative process follows, focusing on enhancing and tuning the model to address any identified shortcomings. It is important to highlight that the "Cornell Movie Corpus" proves instrumental as a training dataset for natural language understanding in this particular context. Lastly, the deployed chatbot undergoes continuous monitoring and maintenance to ensure its effectiveness and responsiveness in real-world interactions.

10. Algorithm Design Steps:

The following algorithmic representation provides a high-level, structured view of the process involved in training and deploying a Seq2Seq-based chatbot model.

```
Algorithm: Seq2Seq Chatbot Model Training and Operation
Input: A dataset of conversational pairs
Output: A trained Seq2Seq model capable of generating chatbot responses
Step 1: Procedure Data-Collection
    Collect a diverse dataset of conversation pairs (D)
    Ensure coverage of various topics and sentence structures
  End Procedure
Step 2: Procedure Data-Preprocessing(D)
    For each conversation pair in D
```

```
      Perform sentence tokenization
      Handle special characters
      Form input-output pairs
    End For
    Return pre-processed dataset (D')
  End Procedure
Step 3: Procedure Convert-To-Numerical-Format(D')
    Initialize Tokenizer T with a vocabulary
    For each data point in D'
      Convert text to numerical format using T
      Apply sequence padding
    End For
    Return numerical dataset (D")
  End Procedure
Step 4: Procedure Seq2SeqModel(D")
    Initialize Encoder E and Decoder Dc
    For each numerical input-output pair in D"
      Pass input to Encoder E
        Encoder-Vector ← Last hidden state of E
      Initialize Decoder input with Encoder-Vector
      While not end-of-sequence token
        Output ← Pass Decoder input to Decoder Dc
        Update Decoder input with Output
      End While
      Combine Decoder outputs to form final response
    End For
  End Procedure
Step 5: Train the Seq2Seq model on dataset D" using Seq2SeqModel procedure
Step 6: Deploy the trained model for chatbot response generation
```

Explaining the notations:

- Procedures are defined for each major step of the process.
- The dataset `D` goes through transformations at each step, represented as `D'`, `D"`, etc.
- The Encoder (`E`) and Decoder (`Dc`) are part of the Seq2SeqModel procedure.
- The loop within the Seq2SeqModel procedure represents the sequential nature of the decoding process.

11. Discussion:

In table 2 the first column shows the input given to our Chat-Bot, the second column shows the expected output, and the third column shows the actual output by the Chat-Bot.

When an input is given, it is necessary for the Chat-Bot to get understanding of the Context of the input otherwise the Chat-Bot will struggle to sustain a conversation that resembles human interaction.

Hence the above inputs are given to check whether the Chat-Bot is able to get understanding of the context properly or not. The above inputs can be grouped in such manner:

I. Greetings: "Hi," "How you doing," "Cool"
II. Apology: "Sorry," "I did not do anything"
III. Question: "To whom," "Which Girl"
IV. Negation: "No"
V. Farewell: "Bye"

11.1 Evaluation of Chat-bot

BLEU, ROUGE, BERT Score, and Distinct-1 metrics.

Table 2: Comparison of Expected and Actual Output, Training, Testing and Evaluation of the model is done on “Cornell Movie Corpus” [33]

Input	Expected Output	Actual Output
Hi	Hello!	Hi
How you doing	I’m doing well, thanks!	I am a little man I have a <OUT>that a
Sorry	No problem.	What did you do to them
To whom	To the person in charge.	To a girl here
Which Girl	The one with the red dress.	This boy<OUT>
I did not do anything	It’s okay, don’t worry.	You say one of the<OUT>youremember one of those
No	Alright.	You have heard of<OUT>
Sorry	It’s alright.	What did you do to them
Cool	Awesome!	Yeah Cool
Bye	Goodbye!	Bye

1. BLEU (Bilingual Evaluation Understudy):
   BLEU Metrics, commonly used to do the evaluation of quality of machine-generated text.
   It works by measuring similarity between the expected output and actual output based on n-grams.
   BLEU Score varies between 0 and 1.

2. ROUGE (Recall-Oriented Understudy for Gisting Evaluation):
   ROUGE Metrics are frequently employed to assess the quality of machine-generated text.
   It works by measuring the overlap of n-grams (usually unigrams, bigrams, and trigrams) between the actual output andthe expected output.
   ROUGE scores also range between 0 and 1.

3. BERT Score:
   BERT Score utilizes pre-trained language models such as BERT (Bidirectional Encoder Representations from Transformers) to gauge the similarity between the generated response and the expected output.
   BERT Score ranges between 0 and 1 [28].

4. Distinct-1:
   Distinct-1 evaluates the diversity of generated responses.
   It works by measuring the percentage of unique unigrams (individual words) in the generated responses.
   Higher Distinct-1 values indicate that the chatbot produces more diverse responses and avoids repetitive outputs.
   Input: Hi
   Expected Output: Hello!
   Actual Output: Hi

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.0
- Distinct-1: 1.0

Input: How you doing
Expected Output: I'm doing well, thanks!
Actual Output: I am a little man I have a <OUT> that a

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.1682
- Distinct-1: 1.0

Input: Sorry
Expected Output: No problem.
Actual Output: What did you do to them

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.0488
- Distinct-1: 1.0

Input: To whom
Expected Output: To the person in charge.
Actual Output: To a girl here

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.0510
- Distinct-1: 1.0

Input: Which Girl
Expected Output: The one with the red dress.
Actual Output: This boy<OUT>

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.0161
- Distinct-1: 0.8

Input: I did not do anything
Expected Output: It’s okay, don’t worry.
Actual Output: You say one of the<OUT>you remember one of those

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.0487
- Distinct-1: 0.625

Input: No
Expected Output: Alright.
Actual Output: You have heard of<OUT>

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.0783
- Distinct-1: 0.75

Input: Sorry
Expected Output: It’s alright.
Actual Output: What did you do to them

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.0488
- Distinct-1: 0.875

Input: Cool
Expected Output: Awesome!
Actual Output: Yeah Cool

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.0165
- Distinct-1: 1.0

Input: Bye
Expected Output: Goodbye!
Actual Output: Bye

- BLEU Score: 0.0
- ROUGE Score: 0.0
- BERT Score: 0.0000
- Distinct-1: 1.0

The comparative performance of the chatbot, assessed through metrics such as BLEU, BERT, ROUGE, and Distinct-1, is presented in two distinct figures, denoted as Figure-4(a) and Figure-4(b). Both figures employ the same axis conventions, with the x-axis representing input questions and the y-axis displaying the corresponding scores achieved by our chatbot. In Figure-4(a), the focus lies on BLEU, BERT, and ROUGE metrics, illustrating their respective scores in relation to the input questions. On the other hand, Figure-4(b) specifically emphasizes the Distinct-1 metric, providing a dedicated visualization of its scores across the spectrum of input questions. These figures collectively offer a comprehensive insight into the chatbot's performance as evaluated by these diverse evaluation metrics.

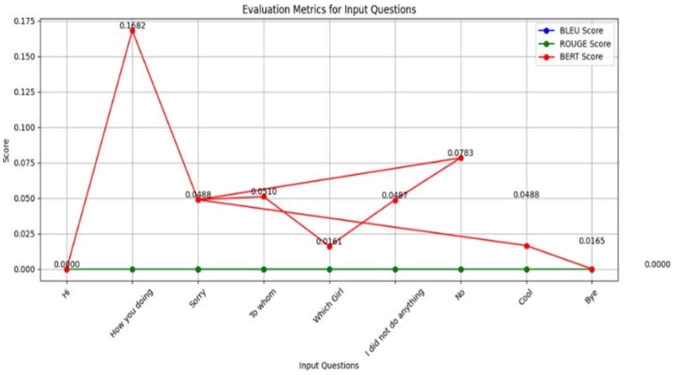

Figure-4(a): Graphical representation of Evaluation Metrics (BLEU, ROUGE, and BERT Score) (BLEU and ROUGE Scores are overlapping with each other)

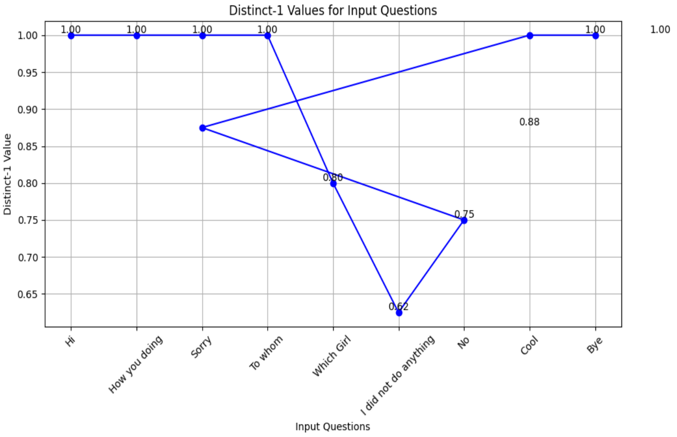

Figure-4(b): Graphical Representation Evaluation Metrics (Distinct-1 Value)

Additional Information on Seq2Seq, Metrics for evaluating the Chat-Bot, and Customization of the Chat-Bot

Seq2Seq

- *Seq2Seq Overview:*
  The Seq2Seq architecture is a deep learning framework crafted for diverse Natural Language Processing (NLP) tasks, including Chat-Bots, Machine Translation, and Text Summarization. It was introduced by Sutskever et al. (Sequence to Sequence Learning with Neural Networks (2014) [23]). Seq2Seq stands out for its proficiency in managing input sequences of varying lengths and its ability to generate output sequences of variable lengths. In the realm of Chat-Bots, the Seq2Seq Model proves valuable for producing variable-length output sequences based on the given input sequence.

- *Seq2Seq Components:*
  Seq2Seq consists of two main components:
  - Encoder:
    The Encoder is tasked with transforming the provided input message (User's message) into the fixed-length context vector. This numerical representation holds the responsibility of capturing the meaning and information embedded in the given input sequence. The Encoder can be constructed using Recurrent Neural Networks (RNN) like LSTM or GRU, or it may adopt modern architectures such as Transformers.
  - Decoder:
    The Decoder utilizes the produced context vector by the Encoder in order to generate output sequence (chatbot's response). Similar to the Encoder, it can be implemented using RNN, LSTM, GRU, or Transformers. The Decoder processes the Context Vector, generating one token at a time while considering the preceding token it has generated.

- *Handling variable Length Sequence:*
  A pivotal benefit of Seq2Seq is its capability to manage variable-length sequences, which holds significant importance for Chat-Bots, given the potential variability in both user input and responses. Special Tokens, such as Start-of-Sequence (SOS) and End-of-Sequence (EOS), are employed to signify the commencement and conclusion of the sequence.

- *Attention Mechanism:*
  To enhance the performance of the Seq2Seq Model, it is frequently augmented with the Attention Mechanism. The attention mechanism was introduced by Bahdanau et al. ("Neural Machine Translation by Jointly Learning to Align and Translate" (2014) [24]). This mechanism empowers the model to focus on different segments of the input sequence while generating individual tokens of the output sequence, thereby contributing to improved context preservation.

- *Fine-Tuning and Transfer Learning:*
  To enhance Chat-Bot performance, training can be conducted on domain-specific data. Transfer learning from pre-trained language models is another strategy employed to bolster performance. Pre-trained models such as BERT can be fine-tuned for chatbot-specific tasks, as demonstrated in the work of Devlin et al [25].

Metrics

BLEU (Bilingual Evaluation Understudy):
The BLEU metric is employed to assess the quality of machine-generated text by measuring the similarity between the machine-generated text and the target sentence (reference sentence). This assessment is based on n-grams, which are contiguous sequences of n words [26].
Mathematical Formulation:
BLEU is calculated in several steps:

- Precision Calculation(P_n):
  Calculate the Precision for each n_gram (like unigrams, bigrams etc.) in the generated text compared to the target sentence (reference sentence).
  Precision quantifies the proportion of n-grams in the generated sentence that are also found in the target

sentence (reference sentence).

$$Precision = \frac{Number\ of\ corrected\ Predicted\ words}{Number\ of\ Predicted\ Words} \quad (1)$$

The model's precision can be heightened by producing fewer words in the output sentence, incentivizing the model to generate shorter outputs. Consequently, there is a need to penalize shorter generated sentences.

- Brevity Penalty (BP):
  Penalize shorter generated text compared to the generated sentence (reference sentence). This accounts for situations where the generated sentence is too short.

$$Brevity\ Penalty = \begin{cases} 1 & if\ \ c > r \\ e^{(1-r/c)} & if\ c \le r \end{cases} \quad (2)$$

Here, c represents the length of the predicted sentence, and r represents the length of the target sentence.

- BLEU Score Calculation:
  Calculate the BLEU score as geometric mean of the precision scores, adjusted by the brevity penalty.
  The BLEU score is calculated as follows:

$$BLEU = BP * \prod_{n=1}^{N} (p_n{}^{w_n}) \quad (3)$$

where is the w_n is the uniform weights, and p_n is the precision of the n-Grams.

ROUGE (Recall-Oriented Understudy for Gisting Evaluation):
The ROUGE metric is employed to assess the quality of machine-generated text.
It evaluates the overlap of n-grams between the generated sentence and the target sentence (reference sentence). [30]

- *Mathematical Formulation:*
  ROUGE is computed by comparing the count of overlapping n-grams in the generated sentence with those in the target sentence (reference sentence). Various ROUGE variants, such as ROUGE-N, ROUGE-L, ROUGE-W, exist, each emphasizing different aspects of text similarity.

For ROUGE-N (unigrams, bigrams, etc.), the formula for precision and recall is similar to BLEU Metric, but it focuses on recall (the proportion of reference n-grams found in machine generated sentence) and precision (the proportion of machine generated n-grams found in target sentence (reference sentence)).

BERT Score:
BERT Score uses pre-trained models like BERT to measure the similarity between the generated sentence and the target sentence (reference sentence). It computes contextual embeddings for both texts and calculates a similarity score [31]

- *Mathematical Formulation:*
  BERT Score relies on cosine similarity between contextual embeddings of the generated and target (reference sentence). Given representations G and T for the generated and target sentence, respectively, the BERT Score (F1 score) is calculated as follows:

$$BERT\ Score = F1 = \frac{2 * precision * recall}{precision + recall} \quad (4)$$

where:

- Precision = cosine similarity (G, T)
- Recall = cosine similarity (T, G)
- Cosine similarity assesses the cosine of the angle between two vectors, providing a measure of their similarity.

Distinct-1:
Distinct-1 is a metric used to evaluate the diversity of generated sentence. It measures the percentage of unique unigrams (individual words) in the generated sentence [21].

Mathematical Formulation:

- Distinct-1 is computed by dividing the count of unique unigrams (distinct words) in the generated sentence by the total number of unigrams in the generated sentence.
- Distinct-1 $= \frac{Number\ of\ Unique\ Unigrams}{Total\ Number\ of\ Unigrams}$ (5)
- This metric provides insight into the diversity of words used in the generated text.

Customization

Yes, chatbots can be designed for a specific industry or domain. This involves training the chatbot with domain-specific corpora so that it chatbot can generate the correct response.

Domain-specific Knowledge Base:

- *Data Collection:*
  Gather the domain-specific data, which includes documents, reports or any other relevant information. For example, in the healthcare industry the relevant information can be patient health records, in the environment domain AQI Index can be relevant information.
- *Knowledge Extraction:*
  Employ NLP techniques, to extract the important information out of the collected data. Tasks such as Named Entity Recognition (NER) can identify key entities, while text summarization can condense lengthy documents [2].
- *Knowledge representation:*
  Represent the extracted information in a structured format, so that the chatbot can process it in a faster way.

Knowledge graphs, ontologies, or databases can be used to represent relationships between entities and concepts [2].

Training Data Preparation:

- *Dialogue Data Collection:*
  If the chatbot will be engaged in dialogues, then collect domain-specific dialogues, which should include the different types of queries and appropriate responses. For example, in the banking industry, the chatbot should be trained on different types of queries like "How to open a savings account", "How to close an account", "How to withdraw money", and so on.
- *Annotation and Labelling:*
  Annotate the dialogue dataset to indicate the intent of user queries and categorize them into relevant topics or actions. For instance, in customer support, queries might be labelled as "Billing Inquiry" or "Technical Support Request" [31].

Customized NLP Models:

- *Fine-tuning Pre-trained Models:*
  Use pre-trained models like BERT, GPT or domain-specific embeddings and fine-tune them on the domain-specific data. Fine-tune adapts the Model to understand the different terminology of the data.
- *Domain-specific Intent Recognition:*
  Train intent recognition models that can identify user intents specific to your domain. Tools like Dialog flow or Rasa can be used to build custom intent recognition models [32].

Domain-Specific Response:

- *Response Generation:*
  Customize the chatbot response according to the domain specific. Response has to be contextual, and accurate. This might involve creating templates for common responses and dynamically filling in details based on user queries.
- *Error-Handling:*
  Error handling makes the chatbot more robust in nature, if the given user's queries are ambiguous or not understood, the chatbot should provide clarification or guide the user appropriately.

Domain-specific Features:

- *Functionalities and Features:*
  Designing and developing features that carter the specific domain.
- *Integration:*
  Integrate the chatbot with domain-specific APIs or databases to fetch real-time data.

Continuous Learning and Improvement:

- *User Feedback Loop:*
  Implement a mechanism for collecting and analysing user feedback continuously. This feedback loop helps in identifying issues, improving responses, and adapting to changing user needs.
- *Monitoring and Analytics:*
  Monitor the chatbot's performance using analytics tools. Track metrics like user satisfaction, response accuracy, and response time to measure how well the chatbot is meeting domain-specific goals.

12. Conclusions

Humans are prone to do mistakes, on the other hand, a computer program cannot do a mistake, minimal involvement of humans in the work is the ultimate goal of this world, Chat Bot is the biggest example of how fast the world is changing, Chat Bot can provide messaging services, Chat Bot can assist, Chat Bot can act as your friend, the applications of Chat Bots are limitless. Before building a Chat Bot it's necessary to decide for what purpose of build the Chat Bot, as building a Chat Bot is a Computational Expensive process, building a Chat Bot which is knowing a particular topic or domain is easy, but building a Chat Bot of Open-Domain is very difficult. That's why it's difficult to build a Chat Bot which can perform all the tasks such as Multimedia Generation, Text Generation, Report Generation and so on. In our project we have built a model using Seq2Seq Model with the Attention Mechanism.

13. Future Work

The model is trained on the movie corpus which only had 20, 000 conversations in it. For future our model can be trained on annotated corpora, parallel corpora and so on, in order to achieve the generalization. By doing this our model will become more informative and will be able to answer more accurately.

References

[1] Alireza Sadeghi Milani, Aaron Cecil-Xavier, Avinash Gupta, J. Cecil & Shelia Kennison (2022) A Systematic Review of Human–Computer Interaction (HCI) Research in Medical and Other Engineering Fields, International Journal of Human–Computer Interaction, DOI: 10.1080/10447318.2022.2116530

[2] Khurana, D., Koli, A., Khatter, K. *et al.* Natural language processing: state of the art, current trends and challenges. *Multimed Tools Appl* **82**, 3713–3744 (2023). https://doi.org/10.1007/s11042-022-13428-4

[3] Adamopoulou, E., Moussiades, L. (2020). An Overview of Chatbot Technology. In: Maglogiannis, I., Iliadis, L., Pimenidis, E. (eds) Artificial Intelligence Applications and Innovations. AIAI 2020. IFIP Advances in Information and Communication Technology, vol 584. Springer, Cham. https://doi.org/10.1007/978-3-030-49186-4_31

[4] Turing, A.M.: Computing machinery and intelligence. Mind 59, 433–460 (1950). https://doi.org/10.1093/mind/LIX.236.433

[5] Brandtzaeg, P.B., Følstad, A.: Why people use chatbots. In: Kompatsiaris, I., et al. (eds.) Internet Science, pp. 377–392. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70284-1_30

[6] Colby, K.M., Weber, S., Hilf, F.D.: Artificial paranoia. Artif. Intell. 2, 1–25 (1971). https:// doi.org/10.1016/0004-3702(71)90002-6

[7] Wallace, R.S.: The anatomy of A.L.I.C.E. In: Epstein, R., Roberts, G., Beber, G. (eds.) Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer, pp. 181–210. Springer, Cham (2009). https://doi.org/10.1007/978-1-4020-6710- 5_13

[8] Marietto, M., et al.: Artificial intelligence markup language: a brief tutorial. Int. J. Comput. Sci. Eng. Surv. 4 (2013). https://doi.org/10.5121/ijcses.2013.4301

[9] Molnár, G., Zoltán, S.: The role of chatbots in formal education. Presented at the 15 September 2018

[10] Colace, F., De Santo, M., Lombardi, M., Pascale, F., Pietrosanto, A., Lemma, S.: Chatbot for e-learning: a case of study. Int. J. Mech. Eng. Robot. Res. 7, 528–533 (2018). https://doi.org/ 10.18178/ijmerr.7.5.528-533

[11] da Costa, P.C.F.: Conversing with personal digital assistants: on gender and artificial intelligence. J. Sci. Technol.Arts 10, 59–72 (2018). https://doi.org/10.7559/citarj.v10i3.563 22. Xu, A., Liu, Z., Guo, Y., Sinha, V., Akkiraju, R.:A new chatbot for customer service on social media. In: Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 3506–3510. ACM, New York (2017)

[12] Følstad, A., Nordheim, C.B., Bjørkli, C.A.: What makes users trust a chatbot for customer service? An exploratory interview study. In: Bodrunova, S.S. (ed.) INSCI 2018. LNCS, vol. 11193, pp. 194–208. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01437-7_16 24. Go, E., Sundar, S.S.: Humanizing chatbots: the effects of visual, identity and conversational cues on humanness perceptions. Comput. Hum. Behav. 97, 304–316 (2019). https://doi.org/ 10.1016/j.chb.2019.01.020

[13] Sannon, S., Stoll, B., DiFranzo, D., Jung, M., Bazarova, N.N.: How personification and interactivity influence stress-related disclosures to conversational agents. In: Companion of the 2018 ACM Conference on Computer Supported Cooperative Work and Social Computing, pp. 285–288. ACM, New York (2018)

[14] Fernandes, A.: NLP, NLU, NLG and how Chatbots work. https://chatbotslife.com/nlp-nlunlg-and-how-chatbots-work-dd7861dfc9df

[15] Ramesh, K., Ravishankaran, S., Joshi, A., Chandrasekaran, K.: A survey of design techniques for conversationalagents. In: Kaushik, S., Gupta, D., Kharb, L., Chahal, D. (eds.) ICICCT 2017. CCIS, vol. 750, pp. 336–350. Springer, Singapore (2017). https://doi.org/10.1007/978- 981-10-6544-6_31

[16] Akma, N., Hafiz, M., Zainal, A., Fairuz, M., Adnan, Z.: Review of chatbots design techniques. Int. J. Comput. Appl. 181, 7–10 (2018). https://doi.org/10.5120/ijca2018917606

[17] Artificial Intelligence Scripting Language - RiveScript.com. https://www.rivescript.com/ 32. Jung, S.: Semantic vector learning for natural language understanding. Comput. Speech Lang. 56, 130–145 (2019). https://doi.org/10.1016/j.csl.2018.12.008

[18] “Chatbot Market Size, Share, Trends | CAGR of 23.91%,” Market.us, Nov. 02, 2023. [Online]. Available: https://market.us/report/chatbot-market/#overview Presented at the (2018)

[19] Nimavat, K., Champaneria, T.: Chatbots: an overview types, architecture, tools and future possibilities. Int. J. Sci.Res. Dev. 5, 1019–1024 (2017)

[20] Kucherbaev, P., Bozzon, A., Houben, G.-J.: Human-aided bots. IEEE Internet Comput. 22, 36–43 (2018). https://doi.org/10.1109/MIC.2018.252095348

[21] Li, J., Galley, M., Brockett, C., Gao, J., & Dolan, B. (2015). A Diversity-Promoting Objective Function for Neural Conversation Models. *ArXiv*. /abs/1510.03055

[22] Balaganesh Bojarajulu, Sarvesh Tanwar, and Thipendra Pal Singh. Parametric and Non-parametric Analysis on MAOA-based Intelligent IoT-BOTNET Attack Detection Model [J]. Int J Performability Eng, 2022, 18(10): 741-750

[23] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. arXiv:1409.3215. https://doi.org/10.48550/arXiv.1409.3215

[24] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural Machine Translation by Jointly Learning to Align and Translate. arXiv:1409.0473. https://doi.org/10.48550/arXiv.1409.0473

[25] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). BERT: Bidirectional Encoder Representations from Transformers. arXiv:1810.04805. https://doi.org/10.48550/arXiv.1810.04805

[26] Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002). BLEU: a Method for Automatic Evaluation of Machine Translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL) https://doi.org/10.3115/1073083.1073135

[27] Lin, C. Y. (2004). ROUGE: A Package for Automatic Evaluation of Summaries. In Text Summarization Branches Out. https://aclanthology.org/W04-1013

[28] Zhang, T., Zhao, H., & LeCun, Y. (2018). BLEU is Not a Good Metric for Legal Text Generation: The (Lack of) Correlation Between BLEU and Human Judgments. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)

[29] Li, J., Monroe, W., & Jurafsky, D. (2016). A Simple, Fast Diverse Decoding Algorithm for Neural Generation. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL) https://doi.org/10.48550/arXiv.1611.08562

[30] Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In Proceedings of COLING (ACL) https://aclanthology.org/C14-1220

[31] Tur, G., & De Mori, R. (2011). Spoken Language Understanding: Systems for Extracting Semantic Information from Speech. John Wiley & Sons.

[32] Open source conversational AI. (2022, October 6). Retrieved from https://rasa.community/#:~:text=Rasa%20uses%20a%20composable%20set,and%20scale%20sophisticated%20conversational%20AI. (Accessed from Dehradun, India on December 01, 23 at 16:54

[33] Niculescu-Mizil, C., & Lee, L, (2011). “Cornell Movie Dialogs Corpus”. Cornell University. https://www.cs.cornell.edu/~cristian/Cornell_Movie_Dialogs_Corpus.html