Decision Analytics Journal 9 (2023) 100359

Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www elsevier com/locate/dajour

An intelligent chatbot design and implementation model using long short-term memory with recurrent neural networks and attention mechanism

Prakash Choudhary [a,*], Sumit Chauhan [b]

[a] Department of Computer Science and Engineering, Central University of Rajasthan, Ajmer, 305817, Rajasthan, India
[b] National Institute of Technology Hamirpur, Hamirpur, 177005, Himachal Pradesh, India

ARTICLE INFO

Keywords:
Deep learning
Chatbot
Natural language processing
Long short-term memory
Recurrent neural network

ABSTRACT

Assistant Conversational Agents (chatbots) are becoming increasingly complex and sophisticated since they are enabled to understand and respond to human language. Chatbots have varying organizational functions, from customer support to personal assistance, and have dominated the digital market as they are efficient and effective. Chatbots can collect valuable information from customers through interaction which can result in a better customer experience, but an inappropriate or inadequate response can also lead to the loss of customers. Therefore, the chatbots must be intelligent enough to provide the most efficient and effective response to positively impact the customers and add value to the business and organization. Over the years, several streams of research have been conducted, and multiple chatbots have been created, but they have their limitation, which has been discussed in this article. We suggest a method to upgrade the performance of the chatbots so that they can respond to the user query with better accuracy. The proposed model shows the implementation of a chatbot using Long Short-Term Memory (LSTM), attention mechanism, Bag of Words (BOW), and beam search decoding. The sequence-to-sequence (Seq2Seq) architecture with an LSTM encoder and decoder has been used. The dialog dataset is preferred to train and test the model, and the BleuN algorithm is used to determine the chatbot's accuracy.

1 Introduction

A chatbot is a set of instructions (program) whose prime objective is to simulate human-like interactions. These chatbots allow users to communicate with machines using natural languages such as English. Chatbots are dominating the digital world and are widely used by profit and non-profit organizations. Because of wide application, each Firm requires chatbots as per their business category and the market they are dealing with. Chatbots can be used in converting website traffic, generating more qualified leads, automating business processes, elevating customer services, and a lot more [1,2]. Chatbots are the key to improve the customer experience as they are capable of handling multiple users on the same instance at a constant pace. These chatbots are capable enough to form an appealing conversation by utilizing large vocabularies and a wide range of topics. Some of the major advantages of using the chatbot are, that it is available around the clock, can provide useful customer insights, better customer satisfaction, and a lot more. Organizations are now relying on chatbots and automated calls as they are quite effective than humans, humans can respond to very limited queries as compared to chatbots which can address mass customers simultaneously. Instead of all these advantages, there are some limitations such as a lack of understanding of the conversational context, they are unable to remember the long conversation which leads to an inappropriate response. A negative response can leave a bad impression on the customer and that customer may not come back, it can cause a loss of the customer. So, it is important to improve the efficiency and accuracy of the chatbot to ensure a better customer relationship and customer retention.

Over the years different models and enhancement techniques have been developed, Nowadays deep learning is one of the best techniques for natural language processing, sequence-to-sequence (Seq2Seq) architecture [3] is a deep learning model that is widely used for implementing chatbot like applications. Attention mechanisms and vectorization techniques are commonly used as an enhancement techniques to support the deep learning model such as Seq2Seq. These enhancement techniques not only improve the performance of the chatbot but also make sure that the responses are relevant and meaningful. Numerous researches have been conducted to push the boundaries of recurrent neural networks and Seq2Seq architecture. Seq2Seq architecture can be used to train any kind of natural language data that has paired conversations. In [4,5] authors proposed different types of attention mechanisms that are available and can be used with Seq2Seq architecture.

* Corresponding author.
E-mail address: prakash.choudhary@curaj.ac.in (P. Choudhary).

https://doi.org/10.1016/j.dajour.2023.100359
Received 3 May 2023; Received in revised form 10 October 2023; Accepted 5 November 2023
Available online 7 November 2023

The performance of chatbots has been a concern for most of the authors as it directly depends on the customer's contentment and the organization's face value. Multiple evaluation metrics are available using which the performance can be evaluated. It completely depends on the organization or author which metrics are to be used as per their implementation, as a standard most of the authors are using Bilingual Evaluation Understudy (BLEU) score as evaluation metrics [6]. The approach discussed in [7,8] bleu-score is used for the performance evaluations and it can be noted that the bleu-score is moderate and can be further improved by using appropriate models and techniques that are available. The goal of this proposed method is to improve the efficiency of the chatbot by using Long Short-Term Memory (LSTMs), Seq2Seq architecture, and Bahdanau attention mechanism.

The rest of the paper is structured as follows. In Section 2, related work has been discussed such as the background of chatbots, and a comparative analysis of existing chatbots are presented in brief. Section 3 is about the dataset selection, data preprocessing, and the methodology used in the proposed work. In Section 4, experimentation and results are shared along with final hyper-parameters, and a comparative study is conducted by referencing methods proposed earlier. Conclusion and future possibilities are proposed in section VI to help the researchers with the limitations and objectives so that they can carry it forward.

2 Related work

Technological advancement leads to the hike in the development of chatbot in the overall market ranging from shopping to finance. Every commercial entity is using the chatbot either directly or indirectly. Because of the wide applications of a chatbot has become a hot topic for researchers and numerous researches have been conducted with their own strengths and limitations. Most of the research in the field of natural language processing is motivated by the possibility of improvement in terms of accuracy. This section highlights the background of chatbot like categorization in terms of interaction, implementation, and domain along with the table for a comparative analysis of the existing chatbots which includes the model used for developing chatbot with their strength and limitations.

2.1. Background of chatbots

There is a wide variety of chatbots available and can be categorized based on multiple criteria such as mode of interaction, implementation approach, goal, and knowledge domain [9]. This broad-ranging categorization is shown in Fig. 1. Chatbots can be designed to interact either via text mode or via voice mode. These modes have their strengths and limitations such as, text-based chatbots should be capable enough to understand the shorthand and typo which is very common in social messaging platforms, voice-based chatbots should be capable of understanding the accent as it can vary from person to person.

On the basis of the implementation approach chatbots are broadly categorized as rule-based, retrieval-based, and generative-based. In Rule-based chatbots, the communications are carried out using predefined rules (like if-else statements). The limitation to such types of chatbots is that the input should be restricted as per the predefined rules, there is no artificial intelligence used in these chatbots and hence these chatbots respond accurately to the predefined inputs but they are unable to simulate human-like interactions for any new inputs [10]. In Retrieval-based chatbots, pattern matching, and deep learning techniques are used to find the appropriate responses [11]. These chatbots are data-oriented as the responses are taken from the predefined repository. The limitation of retrieval-based chatbots is they provide predefined responses and do not produce new responses. In Generative chatbots, predefined repositories are used to train the model so that it can generate a new response for the user input but these chatbots are prone to grammatical mistakes because of which the context of the responses may change [12]. The unavailability of real-life data is also a major concern for retrieval-based and generative-based chatbots, both categories require data for the training and if the dataset lacks quality, it will affect the performance of the chatbot. Retrieval-based and generative-based chatbots are often considered intelligent models as they use artificial intelligence and machine learning approaches.

Generally, chatbots are divided into two categories, task and non-task oriented chatbots. Task-oriented chatbots are designed to perform atomic tasks like appointment scheduling, customer support, and product recommendation. These chatbots are designed to have small conversations and end as soon as the task gets completed, while the non-task oriented chatbots are capable of simulating human-like conversations and are designed to hold the conversations for entertainment purposes. Task-oriented chatbots are designed to respond to specific kinds of user input and are limited in functionality like in appointment scheduling they are limited to its functionality of scheduling the appointments by asking some necessary details from the customers [13,14].

Further categorization of chatbots is based on the knowledge domain as open domain and closed domain. Open-domain chatbots are developed to hold conversations like a human and they are generally for entertainment purposes. While closed-domain chatbots are designed specially to respond to queries that belong to a particular field like customer support [15]. Customer support for an IT company is limited to its projects and services and thus it can be considered a closed-domain. Closed-domain chatbots generally keep track of the keywords in user input and respond accordingly such as railway inquiry, it keeps track of time and destination and responds with trains that match these keywords. An open domain chatbot gives the opportunity to converse without any restrictions but in the closed domain, if you ask something different most of the time they give a common error like "Sorry I didn't understand".

The conventional chatbots were not very flexible and they were limited in terms of performance because of the rule-based techniques. They respond to the user input only if it matches any existing rules which are defined using simple if-else rules or switch cases. The conventional chatbots fail to understand the intent of the input given by the user because of which there is always some possibility of irrelevant responses also they throw an error if any new input is given to them [16]. This drawback can be overcome by practicing machine learning or deep learning techniques. Pattern matching or pattern growth techniques can also be used for identifying the intent of a chatbot or any application of natural language processing [17]. Deep learning chatbot implementation first tries to pull out the meaning of the user input and later it finds or generates the appropriate response for the input. In any conversation, we deal with a situation where the user input and response can be of any arbitrary length which can be handled using many-to-many recurrent neural networks. The Seq2Seq architecture uses many-to-many recurrent neural network (RNN) which is one of the reasons to use it in the proposed work.

2.2. Literature survey

The traditional chatbots used rule-based techniques but the upswing of artificial intelligence has introduced many possibilities. Several researchers proposed multiple techniques that can be used to make a chatbot more efficient, some of the popular deep learning techniques are RNNs, LSTMs, extended LSTMs, Bi-Directional RNNs, etc [15].

Chatbot seems to be a voguish word but they have existed since humans started interacting with systems. The first chatbot was presented before the development of the first personal computer in 1966 named Eliza [18] Since then several chatbots have been introduced. Chatbots have always existed in roles from the internet era when there were no smartphones and have evolved in the smartphone era. Nowadays the advancement in artificial intelligence is giving opportunities to researchers to develop chatbots that are able to mimic human conversations. Fig. 2, shows the timeline of chatbots in different eras like the internet era, smartphones era, and Artificial intelligence era.

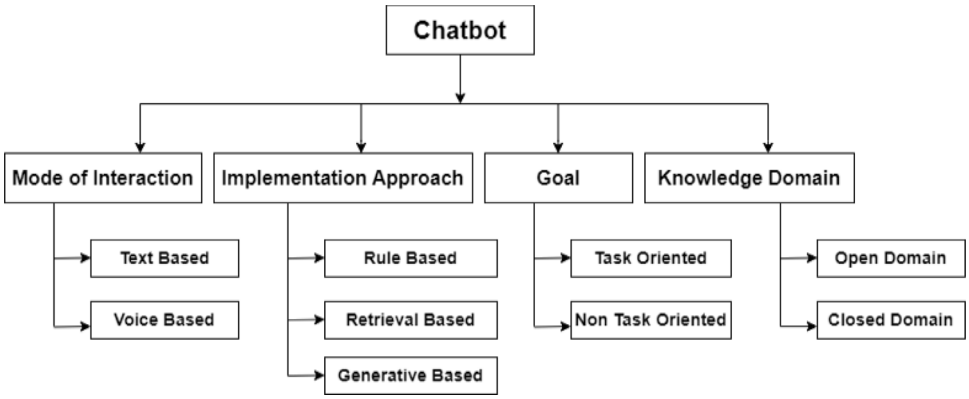

Fig 1 Broad categorization of a chatbot and implementation techniques.

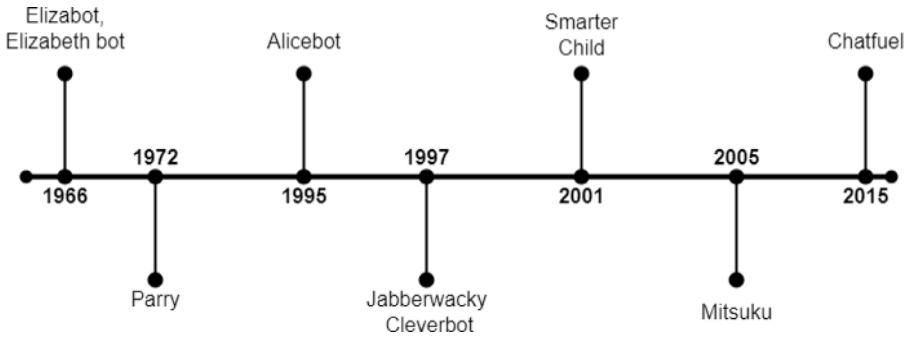

Fig 2 A timeline of Chatbot in different eras.

In Table 1, the Evolution and comparative study of chatbots are conducted. These chatbots were developed in the Internet era. Several other chatbots are developed in the smartphone era such as Siri by Apple, Google Assistant by Google, Cortana by Microsoft, Bixby by Samsung and Alexa by Amazon [19]. All these chatbots that were developed in the smartphone era are generally question-answer systems that respond to user's queries by searching the web.

Recurrent Neural Networks are the most common networks for natural language processing and the chatbot is one of its applications. RNNs belong to the family of Deep Learning Neural networks which also include Convolution neural networks (CNN) and Artificial neural networks (ANN). Like ANN, RNN is also a collection of 3 layers namely the input layer, several hidden layers, and an output layer but the looping constraints on the hidden layers separate RNN, ANN is a simple feed-forward networks with no loop. RNN has a temporal loop on the hidden layers using which it updates the weights at the time of back-propagating to minimize the error while training. This updation of weights are useful for getting a better output but while training the RNN, the Gradient Exploding and Vanishing Problem can emerge when long-term component go dramatically quick to standard 0, making it difficult for the model to get familiar with the connection between transiently distinct events [20]. Some solutions to this vanishing gradient problem are Weight Initialization, Echo State Networks, and LSTMs. The proposed model uses LSTMs to overcome the limitations of RNNs [21].

3 Dataset and methodology

Several kinds of research have been conducted and multiple experiments have been done over the years for implementing the chatbot. Apart from the traditional rule-based techniques, advanced algorithms and techniques such as artificial intelligence are in trend to develop a chatbot. Dataset plays a vital role in training a model using these advanced algorithms and techniques, this section describes the dataset along with deep learning and enhancement techniques used in the proposed work.

3.1. Data-set

Training of a deep learning-based approach always depends on the quality of the dataset used, the better dataset always gives better results if the model utilizes it efficiently. Data-set helps the chatbot in understanding what a person is saying and what to say back as a response. A huge amount of data is required to develop an effective chatbot but the bottleneck is the unavailability of a realistic dataset, quality dataset often leads to effective conversational agents. For natural language processing, several datasets are available like Reddit, Cornell movie dialog corpus, Movie-Dic, Ubuntu dialog corpus, etc [35]. These datasets do not represent real-life conversations and are not enriched with quality.

Firstly, the proposed work has been carried out using the Dialog dataset [36], which resembles real-life conversations. The data is in text format and best suited for chatbot implementation. This dataset contains a single file with 3725 items. The whole dataset is structured in two columns one contains questions and the other columns contains the answers. In most of the pairs, some unnecessary characters and punctuation are present and hold little meaning, thus they have been removed. All this data cleaning and filtering are carried out in data preprocessing.

3.1.1. Data preprocessing

Data preprocessing is an important procedure, it magnifies the dataset quality by encouraging the extrication of meanings from the dataset. It refers to the process of preparing the data and making it suitable for machine learning and artificial intelligence-based models. It can be referred to as data mining to convert the raw data into a meaningful and understandable format. In data preprocessing, the data needs to be cleaned by removing unnecessary characters like punctuations and abbreviations [37]. All the alphabets in the dataset

Table 1
Comparative analysis of existing chatbots.

Chatbot name	Techniques used	Strengths	Limitations
Elizabot, Elizabeth bot [18,22,23]	Rule-based techniques with a script which respond to users input, Elizabeth Bot additionally uses keyword patterns	Elizabot convinced a few people and aided the patient's suffering from intellectual problems. Elizabeth Bot was capable to provide the derivation structure of the sentence using grammatical study	Elizabot was unable to keep the conversation going and failed to understand new words. Elizabeth Bot was unable to split the input and combine the output
Parry [24,25]	Rule-based techniques, pattern matching	Richer control structure and language understanding ability.	Does not utilize grammar in input processing
Alicebot [26]	Pattern matching, Artificial Intelligence Markup Language (AIML) templates, and depth-first search	Holds the conversation with humans using rules of pattern matching	Unable to generate relevant responses
Jabberwacky [27,28]	Contextual pattern matching techniques	No hard-coded rules, it learns from user input	Tries to change the topic and also tries to end the conversation
Cleverbot [29]	Artificial Intelligence algorithm and keyword matching	Responses are not hard-coded, it learns from user input	Unpredictable responses
SmarterChild [30,31]	Natural Language Comprehension	Text Recognition	Unable to hold the conversations
Mitsuku [32,33]	AIML techniques and natural language processing (NLP) using heuristic Techniques	It Remember the personal details of the users like Name, Age, Gender, etc.	Newly learned data needs to be verified then only it can be used
Chatfuel [34]	Rule-based techniques	Allows integration with social media services and Customer Relationship Management (CRM)	Inflexible conversation flows

are converted to lowercase and all the symbols are removed to avoid inconsistency. The steps involved in data preprocessing are shown in Fig. 3.

First, we need data to train the deep learning model, so data should be imported and lines in the data mapped with line ID for identifying the conversation pairs. Step 3 shows, that the sentences involved in one conversational exchange should belong to a list, similarly, lists will be created for all conversations. In step 4, all the questions and answers need to be separated for effective training. Text quality is to be enriched by cleaning the text such as removing punctuation and replacing short-hands [38].

Once text cleaning is done, every word is counted to identify the vocabulary size. Later in step 6, question words and answer words are mapped to unique integers, and a dictionary is created to store these integers. In step 7, tokens like 'PAD', 'EOS', 'OUT', and 'SOS' are added to these dictionaries, in the next step the questions and answers are converted into integers and sorted according to the length of the questions. After all these steps data is appropriate to give as an input to the model for training [39].

3.2. Methodology

The proposed approach is based on LSTMs, Seq2Seq Architecture, Attention Mechanism, Bag of Words (BOW) model, and Beam search decoding. LSTMs with attention mechanisms are not only effective for longer sentences but also for short sentences. This section gives a brief overview of each methodology and also why these methodologies are preferred in the proposed method.

3.2.1. LSTM-RNN

LSTMs were introduced in 1997 by Hochreiter and Schmidhuber [40]. LSTMs are unique RNNs, they have the ability to learn long-term dependencies. RNNs are efficient when the previous information is not required to perform any task as often we need to look ahead to execute any task.

Traditional RNNs are unable to connect the information collected earlier with the recent tasks, in natural language processing the statement made earlier also contributes to the upcoming statements. All RNNs form chains of the recurrent modules but in traditional RNNs the structure is quite simple, each module contains only one interacting layer tanh, as shown in Fig. 4.

Like RNNs, LSTMs also have the same structure but here in each recurrent module instead of only one interacting layer, we have four interacting layers [41] as shown in Fig. 5.

LSTMs utilize gates to include or eliminate information to cell states; these gates are composed of point-wise operations and sigmoid functions [42]. LSTMs mainly take two inputs C_{t-1}, Y_{t-1} from the previous module, and X_t from the current module. The gating mechanism in each cell makes LSTMs different from RNNs.

The equations for the gates and cell vector are:

$$f_t = \sigma\ W_f[Y_{t-1}\ \ X_t] + b_f) \tag{1}$$

$$i_t = \sigma\ W_i[Y_{t-1}\ \ X_t] + b_i) \tag{2}$$

$$\overline{C_t} = tanh\ W_C[Y_{t-1}\ \ X_t] + b_C) \tag{3}$$

$$C_t = f_t\ C_{t-1}) + i_t\overline{C_t} \tag{4}$$

$$o_t = \sigma\ W_o[Y_{t-1}\ \ X_t] + b_o) \tag{5}$$

$$Y_t = o_t tanh\ C_t) \tag{6}$$

Where f_t, i_t, $\overline{C_t}$, C_t, o_t, and Y_t represents forget gate, input gate, cell vector, new cell state, output gate, and LSTMs output respectively. W_* are the input weights and b_* are the bias weights. The Sigmoid() function is used as the gate activator, tanh() function is used as the input and output activator. The role of the cell state is to ensure the flow of information with minimal interaction and it is controlled using forget, input, and output gates.

Unlike RNNs, LSTMs are capable of handling the long-term dependency problem, gradient exploding and Vanishing problem as they have additional controlling knobs in the form of gates and cell states which can control the flow and provide better results. LSTMs are specially designed as they are able to remember the information for a long time [41] which gives us an edge over the traditional RNNs. LSTMs are better as compared to RNNs in almost all tasks, and because of this they are widely used.

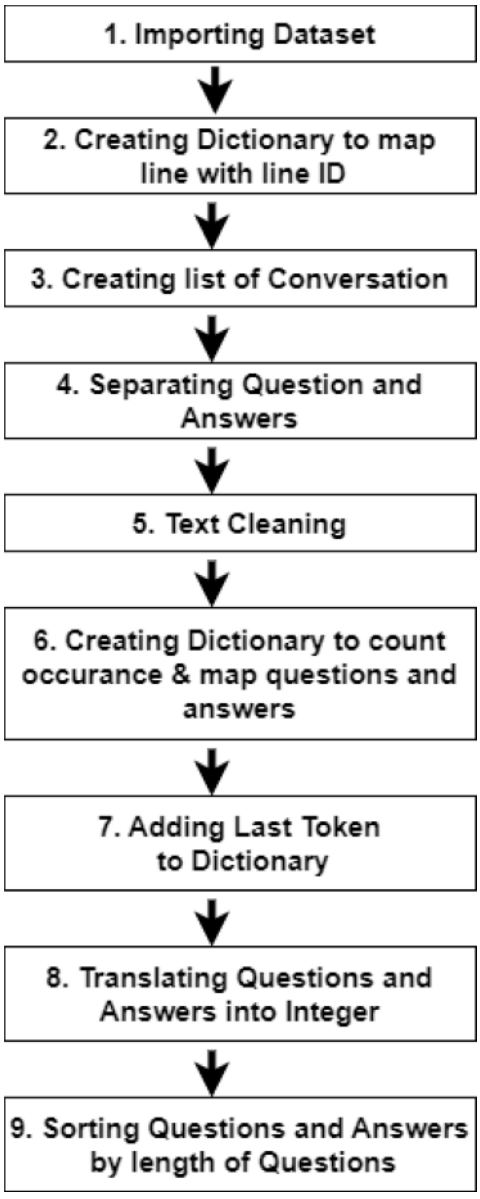

Fig 3 Various steps of data preprocessing.

3.2.2. Sequence to Sequence Seq2Seq) architecture

This architecture was first introduced by Google in 2014, which aims to map the input with the output. This architecture contains two components — encoder and decoder, these components are merged together to form a giant network with wide applications in natural language processing to solve difficult language-related problems. This architecture was introduced to train the models for converting the inputs from one language to another language such as English to Spanish translation, later by periodic modifications the scope of its application increased. This architecture is widely applicable in areas like text summarization, machine translation, developing chatbots, speech recognition, video captioning, and a lot more.

As shown in Fig. 6, the encoder part contains RNN units in a sequence that takes the single word as an input and then forwards it to the next RNN unit and so on. The encoder part can be considered as the stack of RNNs. Later these words are converted to vectors which aim to encapsulate the input fed in the encoder part. The decoder part takes this as input for better predictions. The decoder is also a stack of RNNs that forwards the input to successive RNN units. These RNN units may be of LSTM or GRU (Graphical recurrent units) to enhance the performance [43].

The Seq2Seq architecture has strengths such as mapping of input and output where the length of input and output may differ and it can also process mostly all kinds of NLP data-sets. Seq2Seq architecture falls under supervised learning and it can train the model in less time as compared to reinforcement learning. Reinforcement learning requires more inputs (large datasets) and time to train a model and is most widely used for generative-based chatbots. However, the proposed model is related to retrieval-based chatbots, so Seq2Seq architecture is preferred. This architecture works fine for shorter inputs but fails if longer input is fed because of the limited memory, to overcome this we use LSTMs instead of RNNs. Attention Mechanism is also used to ensure that the predictions made by the model are appropriate by handling the lengthy sentence and making the model robust.

3.2.3. Attention mechanism

An attention mechanism is among the most influencing ideas in natural language processing and deep learning, it has been widely utilized in image captioning and machine translation. The concept of attention mechanism is introduced as an aid to remembering long sentences. In the attention mechanism instead of focusing on the entire statement, we shift our focus to specific parts of the statement which seem to be important [44]. Seq2Seq architecture is not effective when the longer statements are fed as the architecture is not able to remember the longer statements. It forgets the statements that have been processed earlier. This is where the attention mechanism comes into play [45].

In Seq2Seq architecture, we feed the input to the encoder which is processed and converted to a meaningful vector h_n) as shown in Fig. 7. This vector is the input to the decoder and the responses generated are based on this vector. A major challenge with this implementation is the mean vector which is a fixed dimensional component and can store the vectors up to a limit and if we have a variable-length input (like 100 words or more), it will be difficult to vectorize. The attention mechanism handles this by highlighting certain parts of the statements while ignoring the rest of the part. The important part gets vectorized and fed to the decoder to accurately predict the response. While predicting the response, the decoder keeps an eye on the input, and at every prediction of the word, it creates the context vector which contains the weighted sum of all the layers of the encoder. It assigns the weight as per the importance of words to accurately predict the output. This context vector is fed to the decoder layer to predict the new word in response and so on continuously till the complete response is generated [44].

In Fig. 8, the decoder has predicted: "you" (at g_0). According to data training that has been done, it will assign weights as per the importance of the words before predicting "She".

The attention weights a_{ij}) are calculated using softmax function as,

$$a_{ij} = \frac{exp\ e_{ij})}{\sum_{k=1}^{T_x} exp\ e_{ik})}$$

$$e_{ij} = l\ g_{i-1}\ h_j)$$

A context vector C_i) is created as,

$$C_i = \sum_{j=1}^{T_x} a_{ij} h_j$$

Where the output score e_{i_j} is described by function l which is used to capture the alignment among input at j and output at i. T_x is the annotation such as h_0, h_1, h_2, h_3, h_4. The attention weights calculated are a_{1_0}, a_{1_1}, a_{1_2}, a_{1_3}, a_{1_4} are 0.05, 0.2, 0.05, 0.6, and 0.1 respectively. This context vector C_i is now fed to g_1 along with the output of previous decoder layers. This helps the decoder in predicting that the word "sister" means the gender is female, which is a person and the statement is singular. As per these observations, it predicts that "she" is the appropriate response, instead of "sister" the words would be

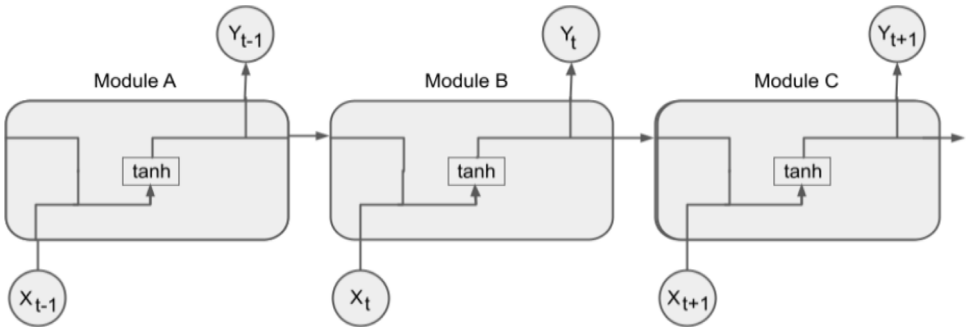

Fig 4 Recurrence of modules in RNN with single tanh layer, redraw of RNN.

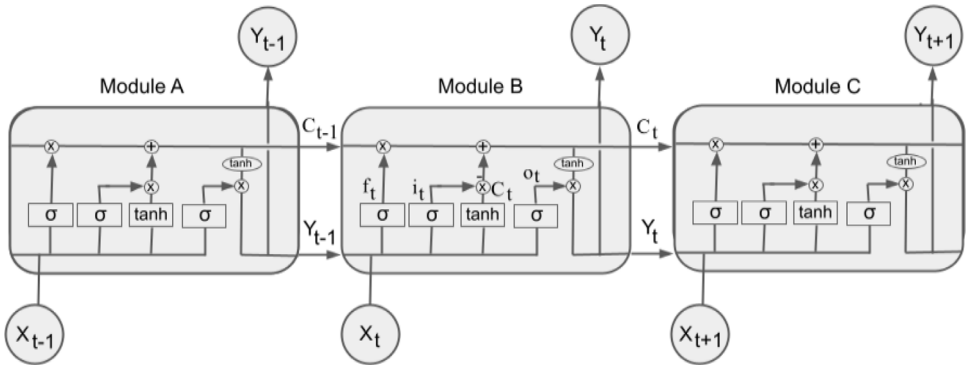

Fig 5 Recurrence of modules in LSTMs with 4 interacting layers, redraw of LSTM.

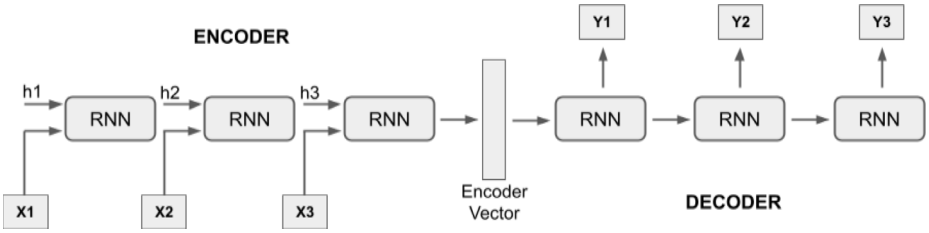

Fig 6 Encoder-Decoder, Sequence to Sequence architecture.

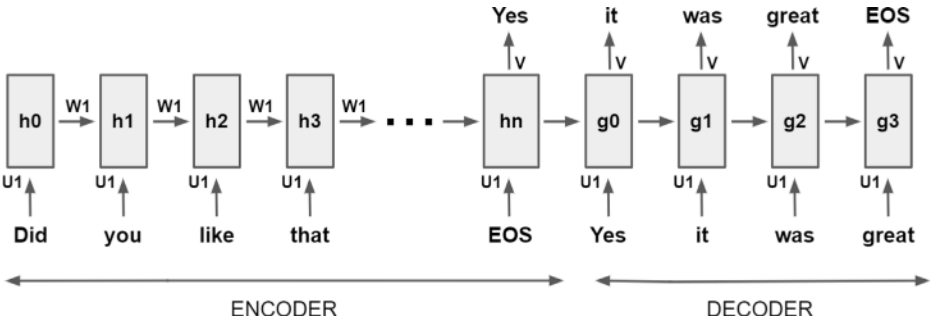

Fig 7 Example of Encoder-Decoder Sequence to Sequence architecture.

"sisters" and it may predict "they". The same process is carried out for the prediction of all words in the response [5]. The mechanism helps the model in paying attention to the specific words of the sentence by initializing the weights, higher attention weights mean that the word is more important and plays an important role in making predictions. It is a popular practice to use an attention mechanism with the seq2seq model.

3.2.4. Bag-of-Words BOW)

Raw data cannot be fed directly to any model for processing and thus it needs to be changed into numbers known as vectors. BOW is a data representation model, frequently used for feature extraction from text files and these features are utilized in deep learning and AI-based algorithms. It creates a list of unique words present in the file. BOW is a model that represents the input text into static vectors by

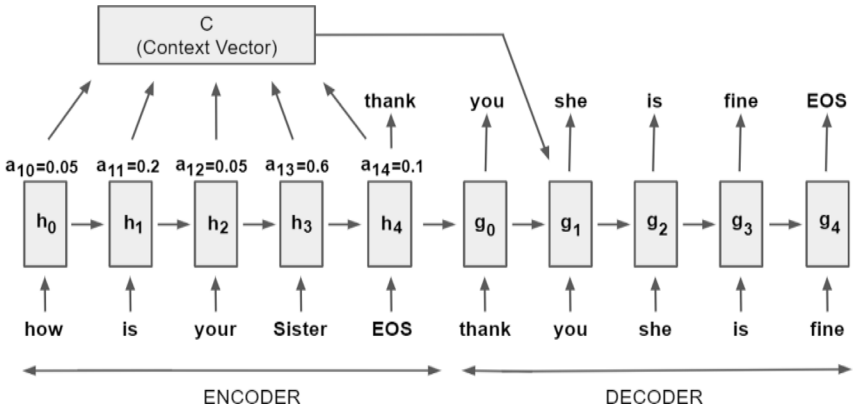

Fig 8 Working of attention mechanism in Encoder-Decoder Sequence to Sequence architecture.

Table 2
Vectorizing the given input sentences.

Sentence	the	cat	saw	rat	was	in	hat	ran	after	seeing
the cat saw the rat	2	1	1	1	0	0	0	0	0	0
the rat was in the hat	2	0	0	1	1	1	1	0	0	0
rat ran after seeing the cat	1	1	0	1	0	0	0	1	1	1

Table 3
Sentence and corresponding Vectors.

Sentence	Vector
the cat saw the rat	[2 1 1 1 0 0 0 0 0 0]
the rat was in the hat	[2 0 0 1 1 1 1 0 0 0]
rat ran after seeing the cat	[1 1 0 1 0 0 0 1 1 1]

counting the existence of every word in the given text. It converts the arbitrary-length text to fixed-length vectors. Firstly we have to define the vocabulary using which we can define the length of the vector [46].

For example, let us have the following statements:

(i) the cat saw the rat.

(ii) the rat was in the hat.

(iii) rat ran after seeing the cat.

In this example, all three sentences are of different lengths and can be converted to fixed-length vectors. To define the vocabulary we should know the words used in the statements, in all the above three sentences a total of 10 words are used like, the cat, rat, was, in, etc. So, we will define the vector length as 10, as the name suggests this vector of length 10 is referred to as a bag with a capacity of 10 units. Now we have vectors of length for all three sentences. Table 2, shows that the occurrence of each word has been counted and vectors are formed for each sentence as shown in Table 3.

These vectors are the input to the model and responses will be predicted as per the data trained. The BOW model is quite straightforward to learn and use. Most of the models work better with fixed-length input and this model can transform the variable-length text into fixed-length vectors. BOW model has its own limitations as the vector size can be large and requires more storage and computational resources when the vocabulary size is large as a result of which the Word2Vec model and Term Frequency-Inverse Document Frequency (TF-IDF) model are common among the developers who use bigger dataset but in this proposed approach we are using a dataset with very limited vocabulary size thus BOW model is fine for this proposed method.

3.2.5. Beam search decoding

Beam search decoding is a technique that generates the sequential translation from left to right, it considers all the possibilities, maintains multiple lists, and returns the best one. While translating the input only those beams are active which have a high probability while the low probability beams are discarded to avoid the overhead [47]. The beam search decoding is introduced to overcome the limitations of greedy decoding, greedy decoding predicts the high-probability word by looking at the input that has been fed to the neural network, it checks the probability for just one word which may not give the right sentence. Whereas the beam search decoding creates beams of words and collectively considers the probability of each beam for better prediction.

Fig 9 Response using Greedy Decoding.

Let us take an example, the input is “Are you in town?” In greedy decoding it first checks the input word “are” and predicts one word with a high probability “yes”, then it checks for the second word “you” and predicts “I” and so on. Let the prediction made by greedy decoding be “Yes I am” as shown in Fig. 9.

Beam search decoding is an upgraded genre of Greedy decoding. Unlike greedy decoding, beam search decoding maintains several streams. It selects the stream with a collective high probability of all words in the stream. As shown in Fig. 10, beam search is a kind of graph search that starts from a node and expands it by looking at the probability. In Fig. 10 a beam search decoding example is shown where it maintains several streams and discards the streams with low probability like “yes why send”. This does not make sense and can be dropped from further predictions if the collective probability is low. The beam with the best collective probability is picked as a response to user input, it may be “yes I am back”. Beam search decoding is one of the best techniques for machine translation which is used with sequence-to-sequence models in NLP as it considers all the possibilities which is not the case with the greedy one. These strengths are utilized in the proposed work.

4 Experimentation and results

In the proposed model, recurrent neural networks are used as deep learning techniques with Seq2Seq architecture. The BOW model is used for the vectorization of the input. To enhance the performance of the model, enhancement techniques such as LSTMs, Bahdanau attention mechanism, and beam search decoding are used. To implement the proposed model Google’s collab is used with Graphical Processing

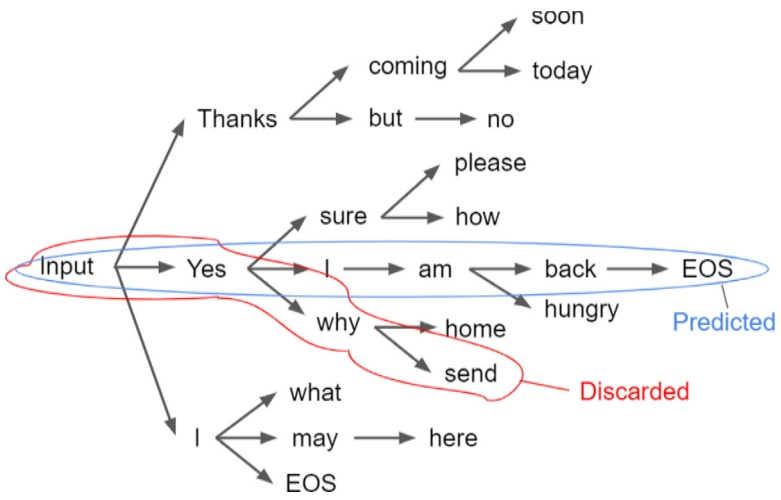

Fig 10 Responses using Beam Search Decoding.

Table 4
Hyper-Parameters used in chatbot implementation.

Parameters	Configurations
Epochs	80
Buffer Size	2980
Batch Size	64
Steps per epochs	46
Learning Rate	0.001
LSTM-RNN Units	1024
Decay Rate	0.9
Encoding Embedding Size	256
Decoding Embedding Size	256

Unit (GPU). The programming language used for the model is Python with libraries such as TensorFlow (version 2.4.1) by Google, Natural Language Toolkit (NLTK), Keras, and NumPy.

4.1. Hyper parameters and training

There is no predefined rule to find out how many epochs are enough or how many recurrent units one should take. Moreover, it is a trial and error-based approach using which we can identify the right set of hyper-parameters.

The parameters changed and tested frequently and the final parameters are shown in Table 4. For training and testing the data is split in an 80:20 ratio. The training is conducted for 80% of the data. The training losses for 80 epochs are shown in Fig. 11, the training loss decreases from 1.8120 to 0.0098 in 80 epochs.

4.2. Results and analysis

Detailed analysis and comparison of proposed models are shared in this section. The responses that are predicted by the proposed method after training the model for 80 Epochs using the "Dialog" dataset are shown in Table 5.

Bilingual Evaluation Understudy (BLEU) score is used to evaluate the predicted responses, in which we compared the predicted response with the reference from the dataset. An exact match results a score of 1.0 and a complete mismatch gives a score of 0.0. Sentence_bleu() function is used from the NLTK library and cumulative bleu scores are calculated for the predicted responses [48]. Bleu N-gram algorithm is used as in the dataset there is one to one relationship. Let the reference is R, the predicted response is O, and W_i is ith n-gram in R or O. So, the N-gram precision P_n is defined as

$$P_n = \frac{\sum_{i=1}^{k} min\ h\ W_i\ O)\ h\ W_i\ R))}{\sum_{i=1}^{k} h\ W_i\ O)}$$

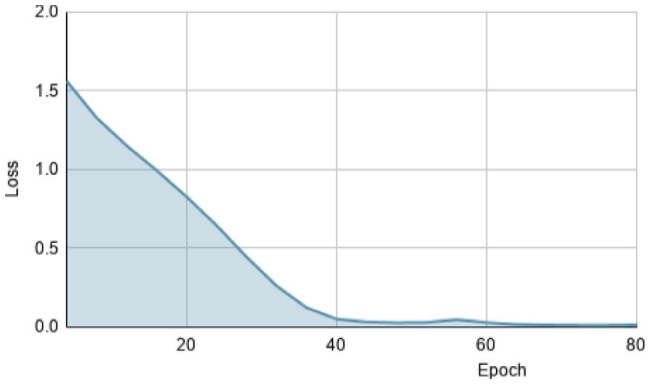

Fig 11 Training losses in 80 epochs.

Table 5
Predicted responses of the proposed method.

Input	Predicted responses
hi how are you doing	i am fine how about yourself
i am pretty good thanks for asking	no problem so how have you been
i have been great what about you	i have been good i am in school right now
what school do you go to	i go to pcc
do you like it there	it is okay it is a really big campus
good luck with school	thanks
i am doing well how about you	never better thanks
how are you doing today	i am doing great what about you
i am absolutely lovely thank you	everything is been good with you
it is an ugly day today	i know i think it may rain

Where $h\ W_i\ O)$ and $h\ W_i\ R)$ are number of W_i in R and O. Typically N-gram favors small sentences, and to handle it brevity penalty (BP) is introduced, which is defined as,

$$BP = \begin{cases} 1 & \text{if } L\ O) \quad L\ R) \\ e^{1-\frac{L\ R)}{L\ O)}} & \text{otherwise} \end{cases}$$

Where L represents the length of reference (R) and predicted response (O). The Bleu score can be defined as,

$$Bleu - N = BP * exp \sum_{n=1}^{N} \frac{1}{N} log P_n)$$

Table 6
Performance comparison using 1-gram, 2-gram, 3-gram, and 4-gram bleu score.

Model	Bleu 1-gram	Bleu 2-gram	Bleu 3-gram	Bleu 4-gram
Responses of CBET [7]	0.714	0.679	0.642	0.608
Responses of DeepQA-based chatbot [7]	0.280	0.262	0.253	0.243
Proposed Method	**0 8833333**	**0 8707107**	**0 8632878**	**0 8537285**

Table 7
Performance comparison using bleu score.

Model	Bleu Score
DeepProbe rewrites without attention [49]	0.326
DeepProbe rewrites with dot attention [49]	0.349
DeepProbe rewrites with general attention [49]	0.388
DeepProbe rewrites with concat attention [49]	0.331
DeepProbe rewrites with tensor attention [49]	0.364
Proposed approach with bahdanau attention	**0 8537285**

The effectiveness of the proposed model is verified using the bleu score. The bleu scores of the proposed approach are compared in Table 6. In the referenced approach, the best bleu score for 10 records has been taken from the set of 500 records. Out of those 10 records, the best one is represented in the table and compared with the average of 10 predictions made by the model in this proposed approach.

In Table 7, the bleu score of the proposed model is compared with Ref. [49], where the bleu score is calculated for each rewrite with different attention mechanisms. In this proposed model we have used the Bahdanau attention mechanism and the bleu score has been calculated as an average of 10 predictions made by the proposed model.

5 Conclusion and future work

In this proposed model we used LSTMs and Seq2Seq architecture with enhancement techniques such as the Bahdanau attention mechanism, BOW model for vectorization, and beam search decoding for response prediction. The chatbot developed is an open domain, a dialog dataset is used and the responses are genuine too. The use of LSTMs, Seq2Seq architecture, and enhancement techniques helped us to achieve the objective of getting better responses with a cumulative Bleu-4 score of 0.8537285. We have compared and analyzed the bleu score of the proposed method with a chatbot for the field of educational technology (CBET), DeepQA-based chatbot, and DeepProbe rewrites and noted that the proposed method is ahead in terms of bleu score and performance.

As future work, this model can be implemented using different datasets such as Cornell movie data corpus or Reddit dataset and can be implemented using Bi-directional LSTM or Extended LSTMs instead of traditional LSTM-RNN. The suggested approach can give better results at the cost of higher computation time. Experimentation can also be carried out to test the possibilities of appropriate hyper-parameters. Some better enhancement techniques can be used to further improve the accuracy of the chatbot, instead of the Bahdanau attention mechanism some other attention mechanism can also be used and in place of a bag of words (BOW), Word2Vec and Term Frequency-Inverse Document Frequency (TF-IDF) can be used and tested to figure out the best one.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] Nuria Haristiani1, Artificial Intelligence (AI) chatbot as language learning medium: An inquiry, J. Phys.: Conf. Ser. 1387 (2019) 012020.

[2] Umasankar Murugesan, Padmavathy Subramanian, Shefali Srivastava, Ashish Dwivedi, A study of artificial intelligence impacts on human resource digitalization in Industry 4.0, Decis. Anal. J. 7 (2023) http://dx.doi.org/10.1016/j.dajour.2023.100249.

[3] Ilya Sutskever, Oriol Vinyals, Quoc V.V. Le, Sequence to sequence learning with neural networks, in: Advances in Neural Information Processing Systems, 2014, pp. 3104–3112.

[4] Xiao mei Yu, et al., An attention mechanism and multi-granularity-based bi-LSTM model for Chinese Q & A system, Soft Comput. 24 (8) (2020) 5831–5845.

[5] Ashish Vaswani, et al., Attention is all you need, 2017, arXiv preprint arXiv: 1706.03762.

[6] Kishore Papineni, et al., Bleu: A method for automatic evaluation of machine translation, in: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, 2002.

[7] Qingtang Liu, et al., CBET: design and evaluation of a domain-specific chatbot for mobile learning, 2019, pp. 1–19, Universal Access in the Information Society.

[8] Youness Saadna, Anouar Abdelhakim Boudhir, Mohamed Ben Ahmed, An analysis of ResNet50 model and rmsprop optimizer for education platform using an intelligent chatbot system, in: Networking, Intelligent Systems and Security: Proceedings of NISS 2021, Springer, Singapore, 2022.

[9] G. Caldarini, S. Jaf, K. McGarry, A literature survey of recent advances in chatbots, Information 13 (2022) 41, http://dx.doi.org/10.3390/info13010041.

[10] E. Adamopoulou, L. Moussiades, An overview of chatbot technology, Artif. Intell. Appl. Innov. 584 (2020) 373–383, http://dx.doi.org/10.1007/978-3-030-49186-4_31.PMCID:PMC7256567.

[11] Yutao Zhu, Jian-Yun Nie, Kun Zhou, Pan Du, Hao Jiang, Zhicheng Dou, Proactive retrieval-based chatbots based on relevant knowledge and goals, in: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '21, Association for Computing Machinery, New York, NY, USA, 2021, pp. 2000–2004, http://dx.doi.org/10.1145/3404835.3463011.

[12] A. Følstad, T. Araujo, E.LC. Law, et al., Future directions for chatbot research: An interdisciplinary research agenda, Computing 103 (2021) 2915–2942, http://dx.doi.org/10.1007/s00607-021-01016-7.

[13] Shafquat Hussain, Omid Ameri Sianaki, Nedal Ababneh, A survey on conversational agents/chatbots classification and design techniques, in: Workshops of the International Conference on Advanced Information Networking and Applications, Springer, Cham, 2019.

[14] Ripon K. Chakrabortty, Mohamed Abdel-Basset, Ahmed M. Ali, A multi-criteria decision analysis model for selecting an optimum customer service chatbot under uncertainty, Decis. Anal. J. 6 (2023) http://dx.doi.org/10.1016/j.dajour.2023.100168.

[15] Chien-Chang Lin, Anna Y.Q. Huang, Stephen J.H. Yang, A review of ai-driven conversational chatbots implementation methodologies and challenges (1999–2022), Sustainability 15 (5) (2023) 4012.

[16] Abdulla Alsharhan, Mostafa Al-Emran, Khaled Shaalan, Chatbot adoption: A multiperspective systematic review and future research agenda, IEEE Trans. Eng. Manage. (2023).

[17] S. Alias, M.S. Sainin, T.Soo. Fun, N. Daut, Identification of conversational intent pattern using pattern-growth technique for academic chatbot, in: R. Chamchong, K. Wong (Eds.), Multi-disciplinary Trends in Artificial Intelligence, MIWAI 2019, in: Lecture Notes in Computer Science, vol. 11909, Springer, Cham, 2019, http://dx.doi.org/10.1007/978-3-030-33709-4_24.

[18] J. Weizenbaum, ELIZA: A computer program for the study of natural language communication between man and machine, Commun. ACM 9 (1) (1966) 36–45.

[19] Alicia Leas Nobles, Eric Caputi, Theodore Zhu, Shu-Hong Strathdee, John Steffanie Ayers, Responses to addiction help-seeking from Alexa, Siri, Google Assistant, Cortana, and Bixby intelligent virtual assistants, npj Digit. Med. 3 (2020) 11, http://dx.doi.org/10.1038/s41746-019-0215-9.

[20] Razvan Pascanu, Tomas Mikolov, Yoshua Bengio, On the difficulty of training recurrent neural networks, in: International Conference on Machine Learning, 2013.

[21] Lokesh Borawar, Ravinder Kaur, ResNet: Solving vanishing gradient in deep networks, in: Proceedings of International Conference on Recent Trends in Computing, ICRTC 2022, Springer Nature Singapore, Singapore, 2023.

[22] J. Epstein, W.D. Klinkenberg, From Eliza to Internet: A brief history of computerized assessment, Comput. Hum. Behav. 17 (3) (2001) 295–314.

[23] Bayan Abu Shawar, Eric Atwell, A Comparison Between Alice and Elizabeth Chatbot Systems, University of Leeds, School of Computing research report 2002.19, 2002.
[24] Mgr Tomáš Zemčík, A brief history of chatbots, 2019, DEStech Transactions on Computer Science and Engineering aicae.
[25] Kenneth Mark Colby, Ten criticisms of parry, ACM SIGART Bull. 48 (1974) 5–9.
[26] Richard S. Wallace, The Anatomy of ALICE. Parsing the Turing Test, Springer, Dordrecht, 2009, 181-210.
[27] About the Jabberwacky AI [Online], Retrieved 19 February 2021 http://www.jabberwacky.com/j2about.
[28] Antonella De Angeli, Rollo Carpenter, Stupid computer abuse and social identities, in: Proc. INTERACT 2005 Workshop Abuse: The Darker Side of Human-Computer Interaction, 2005.
[29] Cleverbot, Cleverbot.com. Retrieved 14 2013.
[30] Eleni Adamopoulou, Lefteris Moussiades, Chatbots: History, technology, and applications, Mach. Learn. Appl. 2 (2020) 100006.
[31] SmartChild, https://www.chatbots.org/chatterbot/smarterchild/. Retrieved 3 2021.
[32] S. Mitsuku Worswick, Chatbot: Mitsuku Retrieved 19 2021 Available from https://www.pandorabots.com/mitsuku/.
[33] R. Higashinaka, et al., Towards an open-domain conversational system fully based on natural language processing, in: Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, 2014.
[34] D. Dumik, Chatfuel, 2015, [23/04/2018]; Available from https://everipedia.org/wiki/chatfuel/.
[35] Fabio Galbusera, et al., Automatic diagnosis of spinal disorders on radiographic images: Leveraging existing unstructured datasets with natural language processing, Global Spine J. 13 (5) (2023) 1257–1266.
[36] Dialog.txt Dataset for Chatbot [Available Online] Retrieved february 10 https://www.kaggle.com/grafstor/simple-dialogs-for-chatbot.
[37] Suha Khalil Assayed, Manar Alkhatib, Khaled Shaalan, Artificial intelligence based chatbot for promoting equality in high school advising, in: 2023 4th International Conference on Intelligent Engineering and Management, ICIEM, IEEE, 2023.
[38] Ashlin Deepa, Sai Sravya Thumati, Sandhya Reyya, An efficient deep learning based chatbot for GRIET, in: 2022 IEEE International Conference on Distributed Computing and Electrical Circuits and Electronics, ICDCECE, IEEE, 2022.
[39] Tattari Jalaja, et al., A behavioral chatbot using encoder-decoder architecture: Humanizing conversations, in: 2022 Second International Conference on Interdisciplinary Cyber Physical Systems, ICPS, IEEE, 2022.
[40] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[41] Christopher Olah, Understanding LSTM networks, 2015.
[42] Suha Khalil Assayed, Manar Alkhatib, Khaled Shaalan, Artificial intelligence based chatbot for promoting equality in high school advising, in: 2023 4th International Conference on Intelligent Engineering and Management, ICIEM, IEEE, 2023.
[43] Q. Zhong, L. Ding, J. Liu, B. Du, D. Tao, E2S2: Encoding-enhanced sequence-to-sequence pretraining for language understanding and generation, 2022, arXiv preprint arXiv:2205.14912.
[44] Minh-Thang Luong, Hieu Pham, Christopher D. Manning, Effective approaches to attention-based neural machine translation, 2015, arXiv preprint arXiv:1508.04025.
[45] Weiqiu You, Simeng Sun, Mohit Iyyer, Hard-coded gaussian attention for neural machine translation, 2020, arXiv preprint arXiv:2005.00742.
[46] Jhou Victor, A simple explanation of the bag-of-words model, https://towardsdatascience.com/a-simple-explanation-of-the-bag-of-words-model-b88fc4f4971, Retrieved 10 2021.
[47] Markus Freitag, Yaser Al-Onaizan, Beam search strategies for neural machine translation, 2017, arXiv preprint arXiv:1702.01806.
[48] A gentle introduction to calculating the BLEU Score for Text in Python, https://machinelearningmastery.com/calculate-bleu-score-for-text-python/ Retrieved 8-March, 2021.
[49] Zi Yin, Keng-hao Chang, Ruofei Zhang, Deepprobe: Information directed sequence understanding and chatbot design via recurrent neural networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017.