Proceedings of the 3rd International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/6/20230936

Performance comparison of representative chatbots based on deep learning

Chuqi Song

356 Alexandra Road, ALEXIS, 159949, Singapore

Song0160@e.ntu.edu.sg

Abstract. Chatbots have always been a hot topic in the field of natural language processing, which aims to train models to understand input content and give cognitively logical answers. With the continuous development of artificial intelligence technology, the performance of chat robots has continuously made breakthroughs, and has been widely used in many fields such as smart home, e-commerce, and automatic consultation. In this article, we introduce the basic techniques and methods of deep learning-based chatbots, including recurrent neural network (RNN), long short-term memory (LSTM), Seq2Seq model, attention mechanism, etc. We also review a chatbot designed using RNN and quality testing result. We finally summarize the existing research issues of the chatbot and discuss its possible future works.

Keywords: Deep learning, Chatbot, Natural language processing, Recurrent Neural Network, Long Short-Term Memory

1. Introduction

With the rapid development of computer in the past few decades, artificial intelligence has become a useful partner of human beings in modern life. There are many significate achievements in the area of artificial intelligence recently and among them, one of the most famous areas is Natural Language Processing. Natural Language Processing, also known as NLP, is the method to make natural human language understandable for computer systems [1]. One of the most widely used application of NLP technique is conversational agent, which can also be called as chatbot. Conversational agent, or chatbot, is a system that allow human beings to communicate with machine with natural human language.

The chat robot refers to a program that automatically completes a dialogue with a user and helps the user to perform operations automatically in a specific situation. Nowadays, there are two main types of chatbots, which is task-oriented chatbot or non-task-oriented chatbot. The former one has a very clear aim of service area, for example, to make a reservation for restaurant [2] or customer service. This type of chatbot is limited to certain topics and is not able to answer arbitrary questions. The latter one does not have a specific aim and area of service and is able to handle all sorts of questions. These models are normally trained by daily life conversations and supposed to be hard to differentiate from real human. This type of chatbot is widely applied for entertainment and accompaniment areas.

The earliest Chatbot research can date back to the famous Turing test, which raised the question of whether machines can think. The first chatbots named ELIZA was developed by Weizenbaum in 1966 [3], which is able to analyze the key words of the input sentence using pattern matching logic and give an output using the hard-coded database with the key words of input. This chatbot mainly relies on

DOI: 10.54254/2755-2721/6/20230936

templates. If what the user said matches the already written template, you get a good reply, otherwise you get some catch-all reply. Subsequently, rule-based chatbots gradually emerged in the late 20th century, among which Parry and Alicebot are representative. However, due to the variety of human languages, relying solely on template technology cannot enumerate all situations, so the development of chatbots was once at a bottleneck. After entering the 21st century, with the development of machine learning technology and the increase of available Internet dialogue materials, data-driven chatbot technology has become more mature and has gradually become the mainstream of research. The famous chatbot implemented by this method is Cleverbot. Ever since its launch date on web, it has held more than 150 million conversations with human beings [4]. However, the performance of this kind of chatbot system is limited as it deeply relies on the size and variety of hard-coded dataset.

With the raise of deep learning these years, the above limitation of chatbot has been improved a lot by enabling the computer to generate the output itself, instead of fully depending on dataset. The chatbot system has the ability to study the pattern of human language with the help of neural network. It can generate original and grammatically correct response with the input sentence. With the help of deep learning, recent chatbots can not only process the written inputs, but pictures and speeches as well. Convolutional neural network can be used to process the images to the language that computer can understand, and speech recognition technique is used to convert the speech into text [5]. Among them, the most representative ones are retrieval-based chatbots and generation-based chatbots. Retrieval-based chatbots use information retrieval technology to match a user's conversation request with a pre-stored dialogue material as a reply. Generation-based chatbots refer to the use of natural language generation technology to automatically reply to user conversation requests.

Focusing on the above aspects, in this paper, we introduce the advanced studies of the chatbots based on deep learning. Specifically, in section 2, the essential techniques and methodology of building a chatbot with deep learning is introduced, including Recurrent Neural Network (RNN), Long Short-Term Memory (LSTM), Seq2Seq model, Attention Mechanism as well as Beam Search. Then, one chatbot designed using RNN will be reviewed. The model and dataset used as well as the quality testing result are included. In the last part of the paper, the current limitations of the chatbot design will be discussed and possible resolutions and future works are proposed.

2. Essential techniques

2.1. Recurrent neural network (RNN)

It is known that Neural Network can be regarded as a black box where if there is enough training data give, for any input X, an expected output Y can be generated.

From the Neural Network model it can be inferred that the inputs are independent from each other and does not necessarily have a relationship. For Recurrent Neural Network (RNN), it is designed to handle the scenario when the inputs have inner-connections, which means, it is not accurate to process each input separately. For instance, when the brain processing one sentence, it is clear that the sentence cannot be segmented to words and be processed independently. The meaning of one word in the sentence largely depends on the words before and behind. Thus, when neural network tries to understand an input sentence and give an output, it is better to refer to other words in the same sentence instead of processing one by one, and RNN is commonly used in this situation.

DOI: 10.54254/2755-2721/6/20230936

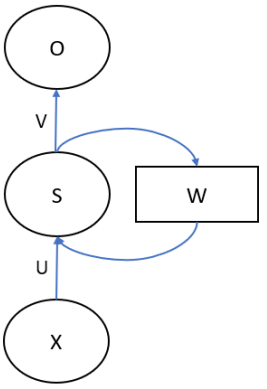

Figure 1. Simple RNN structure.

The structure of a simple RNN is shown in Figure 1, which consists of input layer, hidden layer and output layer, similar as normal neural network models. The difference is at the weight W in hidden layer. if we unfold the simple RNN structure as time manner, shown in Figure 2, it can be inferred that W is the weight of the output of the hidden layer of the last time. Thus, the output of the current hidden layer does not only rely on the current input, the previous output also has an impact on the current hidden layer.

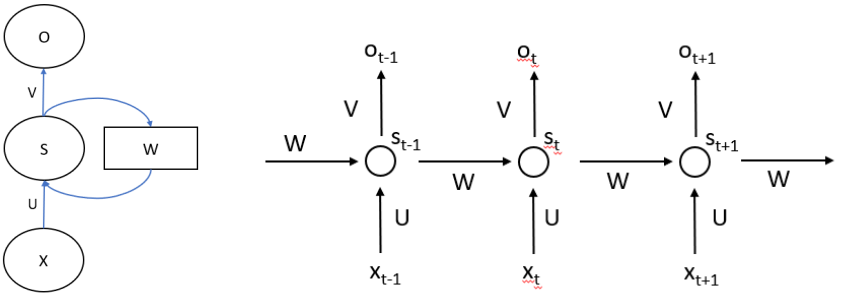

Figure 2. Unfolded RNN structure.

The output of the hidden layer of RNN model can be generated using the following formula:

$$S_t = f(U \cdot X_t + W \cdot S_{t-1}) \quad (1)$$

There are also models that are developed from RNN. Bidirectional Recurrent Neural Network (BiRNN), is consists of two RNN models which are forward and backward, respectively. BiRNN does not only rely on the previous input elements, but also the future ones. Thus, BiRNN has the capability to process the word with the consideration of the before and behind words in a sentence.

DOI: 10.54254/2755-2721/6/20230936

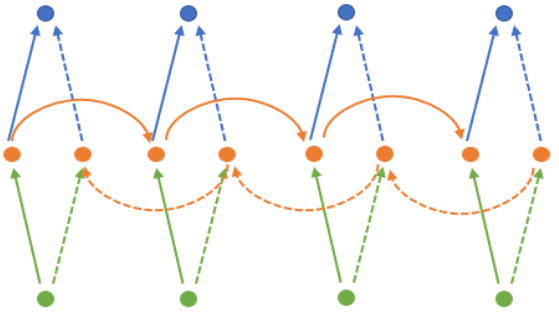

Figure 3. BiRNN structure.

2.2. Long short-term memory (LSTM)

Long Short-term Memory (LSTM) is an advanced version of RNN which was proposed by Hochreiter et al. in 1997 [6]. It is mainly used to resolve the issues of vanishing gradient and exploding gradient which occur during long-sequence training. In short, comparing with simple RNN model, LSTM will have a better performance in the training of longer sequences.

In LSTM, z, z^i, z^f and z^o are generated from the current input x_t and the previous hidden layer output h_{t-1} as below:

$$z = tanh\left(W_{h^{t-1}}^{x^t}\right) \tag{2}$$

$$z^i = \sigma\left(W^i{}_{h^{t-1}}^{x^t}\right) \tag{3}$$

$$z^f = \sigma\left(W^f{}_{h^{t-1}}^{x^t}\right) \tag{4}$$

$$z^o = \sigma\left(W^o{}_{h^{t-1}}^{x^t}\right) \tag{5}$$

z z^i, z^f and z^o are between 0 to 1 as an output of the sigmoid activation function to serve as a state of the gate control. z is the input which is processed by tanh and is between -1 to 1.

The structure of LSTM can be shown as the Figure 4. There are 3 stages for LSTM. The first stage is the forgetting stage, which is to choose whether to forget the previous input. z^f serves are a forgetting gate controller here to control which part of the c^{t-1} needs to be forgotten. The second stage is the memorizing stage which choose the input to memorize. It is controlled by z^i. To add the output of the above mentioned two stages, c^t can be generated. The last stage is the output stage, which is mainly controlled by z^o.

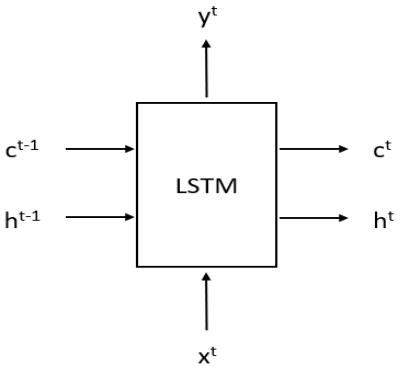

Figure 4. LSTM structure.

DOI: 10.54254/2755-2721/6/20230936

In all, comparing to simple RNN model, LSTM has three more gates to control the information to input, to forget and to output. Instead of memorizing everything like simple RNN model, LSTM can store information selectively.

2.3. Seq2Seq, attention mechanism and beam search

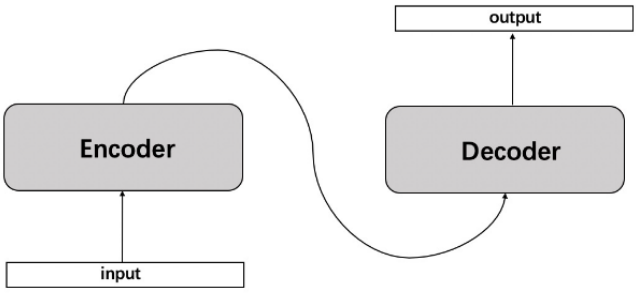

Figure 5. Simple encoder-decoder structure.

Seq2Seq is the model to be used when the size of output is not a fixed value. It is commonly used in NLP tasks, for example, online translation and conversational system. It can be regarded as one usage of the encoder-decoder model [7]. See Figure 5 for the simple encoder-decoder structure. The structure consists of two RNN models. One RNN as encoder and the other one decoder. The encoder RNN can convert the inputs to a vector C with certain length. Then the decoder can convert the vector to a certain array. There are two types of decoding (Figure 6). In the first method, the vector C from encoder will act as the initial state of the decoder RNN. The following computations are irrelevant with C. The other way is for C to be involved in all the computations as well as the initial state of the decoder.

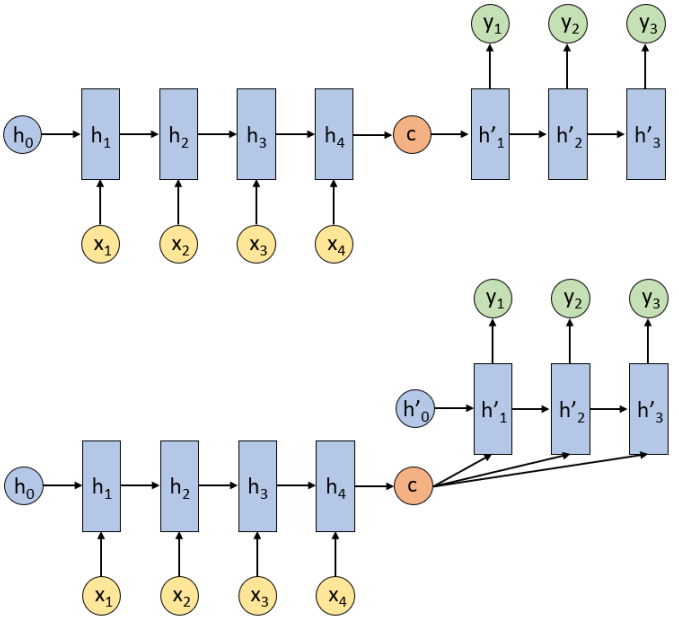

Figure 6. Two encoder-decoder structure for Seq2Seq.

Attention is a special mechanism that can improve the performance of decoder. It was first proposed by Bahdanau et.al in 2015 [8]. In attention mechanism, the vector C generated by encoder will take park in each part of computation in the decoder with a weight. In the decoding process, the weight will be

DOI: 10.54254/2755-2721/6/20230936

adjusted according to the output that it is going to generate. If the hiding stage for encoder is h_j and the length of the input array is T, then the c_i can be calculated using below:

$$c_i = \Sigma_{j=1}^{T} a_{ij} h_j \tag{6}$$

$$a_{ij} = softmax\left(e\left(s_{i-V} h_j\right)\right) \tag{7}$$

If the decoder only uses the previous output as the next input, it will result in the accumulation of error. To resolve the issue, beam search has been introduced. At each cell of decoder, it will select the top K from the vocabulary probability to act as the next input separately, and then select the top K of the K*L (L is the size of the vocabulary list) as the next input. It will select the top 1 from the final output list as the final output.

3. Model, data and experiment

3.1. Dataset

An open-source chatting data [9] has been utilized as the database for this chatbot model. There are over 1.7 million sentences in the format of Q&A in original dataset which is related to children. The next step is data pre-processing and data cleaning. As the chatbot is meant to build for ASD children, improper sentences, like violent and illegal conversations, are removed. After the pre-processing, there are over 400,000 sentences left as the database to serve for model training.

3.2. Build the chatbot

The improved version of structure of RNN, which is BiRNN, is used in this model as the neural network unit. The RNN unit used is also the advance version of RNN unit, which is LSTM. The structure of the LSTM unit used is shown as below (Figure 7).

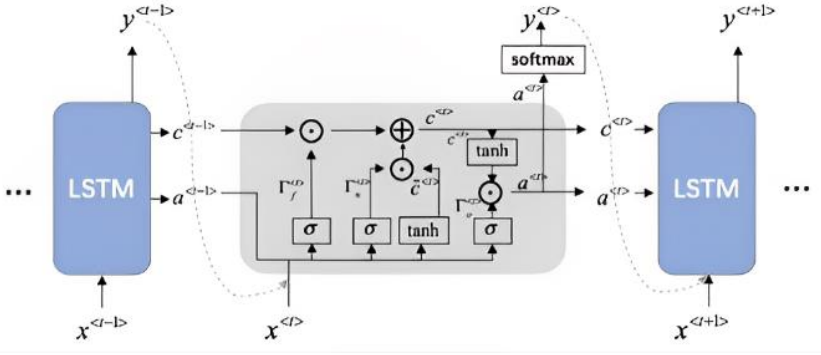

Figure 7. Structure of LSTM framework used.

Seq2Seq model is also used in this chatbot system. There are two RNNs trained in this model and they are serving as encoder and decoder, respectively. Same as the common usage of Seq2Seq in chatbot, this project uses Seq2Seq model encoder to convert the input as one vector and use decoder to output a sequence of words from the vector. The structure of Seq2Seq model used is shown as below (Figure 8):

DOI: 10.54254/2755-2721/6/20230936

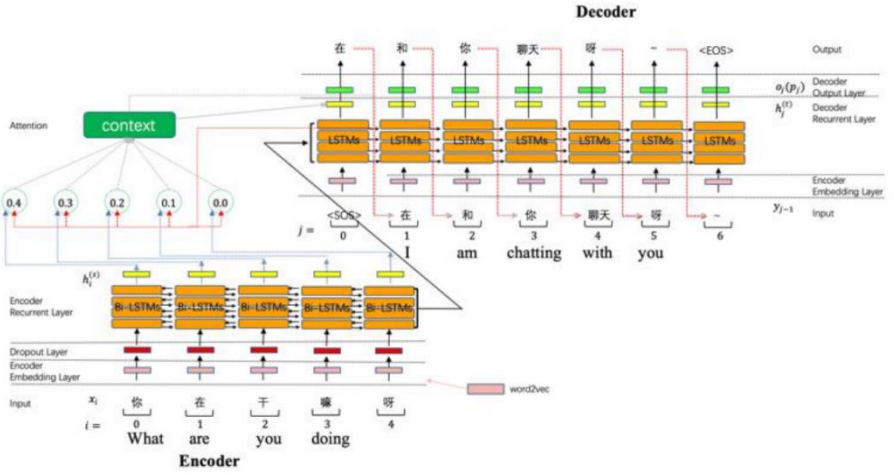

Figure 8. Structure of Seq2Seq model.

Attention mechanism is used in the model to improve the performance. The hidden layer states' attention weight will be learnt by a neural network model and after considering the attention variable, the encoder's hidden layer states can be defined.

Word embedding can help to convert the input more accurately by study the meaning of the words and sentences from a larger dataset. As the words encoded by one-hot vectors might lose inner-sentence semantic connections and make the input meaningless, work embedding technique is introduced to resolve the issue. This project uses Chinese word2vec to process the input vocabulary.

The basic Seq2Seq structure will only select the top 1 vocabulary as the output of the current state and merge them together as the final output. This greedily strategy will lead to error accumulation. Thus, beam search will serve as a solution and it pick up the top K vocabulary from each states as the input of the next state.

3.3. Parameter used in model and training

The dimension of hidden state is set as 256 and word embedding dimension is 300. Four layer BiLSTM is used as encoder and UniLSTM as decoder. The initial learning rate is 0.001 and batch size is 256. It used the Adam optimizer.

3.4. Quality testing

After studying the chatbot evaluation research from Radziwill and Benton [10], the authors designed an evaluation framework based on four dimensions: Humanity, affect, accessibility and performance. As it is hard to give system-generated rating for these dimensions, questionnaires are designed to evaluate the user's experience. The satisfaction mark is from 1 to 100 and the higher the mark given, the better the user's experience is. For the performance measuring, there are two more chatbots collected to compare with the presented chatbot [11][12].

15 adults are invited to finish the evaluation questionnaire. Each of them conducted 20 rounds of conversations with the presented chatbot and the 2 chatbots used for comparison. The result is shown as Figure 9. It can be inferred from the result that the best part that this chatbot works on is affect and accessibility. Consider the fact that this chatbot is specially designed for ASD children, who usually shows less interest in things and reduced language ability, this chatbot indeed performs well in serving it's proposed usage and target users.

DOI: 10.54254/2755-2721/6/20230936

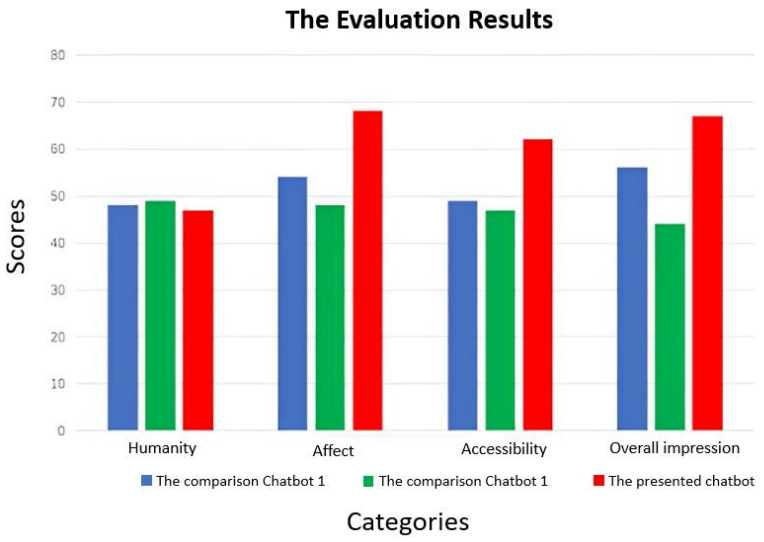

Figure 8. Evaluation questionnaire results.

4. Discussion

Although the current retrieval chatbot has achieved preliminary results, it still faces some serious problems and challenges, mainly including:

(1) *Model reasoning.* Current retrieval chatbots mainly consider context and relevance of candidate responses, but do not perform well when they need to make simple model inferences. The problem of chatbot model reasoning is currently mainly limited by the lack of good data sets for evaluation, making it difficult to conduct research. Current data sets for reasoning are still mainly limited to the field of question answering, such as CommonsenseQA and DROP data sets. Therefore, in order to promote related research on chatbot reasoning problems, a good dataset is essential. The current data-driven algorithms are often incapable of reasoning. How to design a better reasoning model is not only a difficult problem in the research of retrieval chatbots, but also a major problem in the field of natural language processing.

(2) *Multiple rounds of chatting.* When going from a single round of conversational responses to multiple rounds of conversational responses, chatbots performed less than ideally. The main reason is that the inter-sentence relationship in multiple rounds of dialogue is very complex, which may contain various relationships such as succession and transition. Due to the transfer of topics, the content of multiple rounds of dialogue history has no effect on the current reply, or even has a counterproductive effect. Similarly, it is also possible that due to the relationship of succession, the history of multiple rounds of dialogue plays an irreplaceable role in the current response selection. Therefore, how to capture the contextual relationship between multiple rounds of dialogue is crucial.

(3) *Multimodal dialogue.* Current chatbots only consider textual information, but conversations between people often contain more modal information, and this information is critical to the understanding of the conversation. For example: Facial expressions and body language can better aid in the understanding of multiple rounds of conversation. If the facial expression is very ferocious, it means that the speaker is very angry; if the face is smiling, it means that the speaker is very happy. Therefore, when building a chatbot, introducing multi-modal dialogue can better help the understanding of multiple rounds of dialogue, thereby improving the user experience of the chatbot.

5. Conclusion

The chatbot is a program that automatically completes a conversation with a user and helps the user perform actions automatically in specific situations. Thanks to the development of deep learning, especially recurrent neural networks, the performance of chatbot systems has been greatly improved. In

DOI: 10.54254/2755-2721/6/20230936

this paper, we present recent research progress on deep learning-based chatbots. We first introduced the key technologies related to building chatbots, including BiRNN, LSTM, Seq2Seq model, attention mechanism, etc. On the basis of analyzing the performance of representative chatbots, we discuss the remaining problems and future development directions in the field of chatbot research.

References

[1] Jurafsky, D., & Martin, J. H. Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition.
[2] Joshi, C. K., Mi, F., & Faltings, B. (2017). Personalization in goal-oriented dialog. arXiv preprint arXiv:1706.07503.
[3] Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. Communications of the ACM, 9(1), 36-45.
[4] Gilbert, R. L., & Forney, A. (2015). Can avatars pass the Turing test? Intelligent agent perception in a 3D virtual environment. International Journal of Human-Computer Studies, 73, 30-36.
[5] Csaky, R. (2019). Deep learning based chatbot models. arXiv preprint arXiv:1908.08835.
[6] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
[7] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. Advances in neural information processing systems, 27.
[8] Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.
[9] Top. [Online] Available: http://file.rszhang.top/nlp/chat.zip
[10] Radziwill, N. M., & Benton, M. C. (2017). Evaluating quality of chatbots and intelligent conversational agents. arXiv preprint arXiv:1704.04579.
[11] ChatBot. [Online]. Available: https://github.com/1033020837/ChatBot
[12] Why GitHub? [Online]. Available: https://github.com/wudejian789/Attention-Seq2Seq-Chatbot-by-Pytorch1.0.1