International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

A CONDITIONAL GENERATIVE CHATBOT USING TRANSFORMER MODEL

NIVEDITA K M[1], SIDDESH K T[2], KOTRU SWAMY S M[3]

Student, Department Of MCA, BIET, Davangere[1]
Assistant professor, Department Of MCA, BIET, Davangere[2]
Assistant professor, Department Of MCA, BIET, Davangere[3]

ABSTRACT: A Chatbot serves as a communication tool between a human user and a machine to achieve an appropriate answer based on the human input. In more recent approaches, a combination of Natural Language Processing and sequential models are used to build a generative Chatbot. The main challenge of these models is their sequential nature, which leads to less accurate results. To tackle this challenge, in this paper, a novel architecture is proposed using conditional Wasserstein Generative Adversarial Networks and a transformer model for answer generation in Chatbots. While the generator of the proposed model consists of a full transformer model to generate an answer, the discriminator includes only the encoder part of a transformer model followed by a classifier. To the best of our knowledge, this is the first time that a generative Chatbot is proposed using the embedded transformer in both generator and discriminator models. Relying on the parallel computing of the transformer model, the results of the proposed model on the Cornell Movie-Dialog corpus and the Chit-Chat datasets confirm the superiority of the proposed model compared to state-of-the-art alternatives using different evaluation metrics.

Keywords: Generative Chatbot Deep Learning, Conditional generative model, Transformer model Dialog system.

1. INTRODUCTION

Regarding the increasing use of social networks, a new technology called Chatbot has been developed as a tool for human-computer interaction. Chatbots have been widely applied in various fields such as e-commerce [1, 2], education [3], banking and insurance [4], Data collection and management [5], and Health [6]. Chatbots automatically give more attractive answers to users through easy and efficient communications. The key factor in the suitable design of Chatbots is to provide understandable answers to the user [7]. To this end, various approaches have been recently developed to build Chatbots. In general, the approaches of Chatbot development can be classified into two categories: open and closed domain (See Fig. 1). Chatbots with the ability to answer on more than one domain, are called open domains. In contrast, closed domain Chatbots can answer only to questions concerning a particular domain.

Open and closed domains Chatbots can be categorized into Rule-based and Artificial Intelligence (AI)-based Chatbots. Rule-based Chatbots rely on the user's input with a base template. This type of Chatbots choose a predefined answer from a set of answers [8]. However, they are inflexible and limited to some predefined answers. ELIZA and ALICE are based on this approach [9]. AI-based Chatbots are trained to have human-like conversations using a hybrid of NLP and deep learning approaches. More concretely, AI-based Chatbot classified into three categories: Retrieval, Learning, and Generative-based methods. The first category, the retrieval-based method, chooses the most similar answer from a dataset using a functional scoring metric [10]. The second category, the learning-based method, usually learns patterns from training data, containing questions with answers, through deep learning models to generate relevant answers. The last category, the generative-based method, is usually based on the Sequence to Sequence (Seq2Seq) learning models [11-14]. However, the main challenge of these models is their sequential nature, which is led to less accurate results. To tackle this challenge, in recent years, researchers have developed some models based on the transformer models [15-19]. Relying

on the parallel computing as well as the self- attention mechanism of the transformer model, the performance of the models developed using transformers has been improved in comparison with the Seq2Seq models [20]. Although, Learning- based models are not able to generate various answers. To overcome this shortcoming, the generative-based models have been developed to learn the answers distribution.

2. LITERATURE SURVEY

In this section, we briefly review the recent works in four categories: Rule-based, Retrieval -based, Learning-based, and Generative-based.

2.1 Rule-based models

In the rule-based models, the characteristic variables of the input expression are first specified. Then, a predefined answer is provided based on the variables and rules [7]. Rule-based approaches can be divided into two categories: pattern matching methods and standard task-oriented systems [23]. In pattern matching methods, Chatbots match the user's input to the pattern of the rules and select a predetermined answer from the set of answers using pattern matching algorithms. Task-oriented systems guide the user to complete certain items. Since the 1990s, a great deal of research has been conducted into the design of Chatbots based on similar rules for providing services in specific domains. These Chatbots are known as task- oriented modular chat systems that guide the user to perform some structured tasks, such as restaurant and film reservations. Since 1966, the development of the Eliza Chatbot has begun with a pattern-based approach. This Chatbot analyzed the input sentence based on the parsing rules established by the keywords in a sentence [24]. "Pari" adds some influential variables like "fear", "anger" and "distrust" to the more complex rules. These rules have made the conversation more humane-like [25]. ALICE uses Artificial Intelligence Markup Language (AIML), which is the category constituting the unit of knowledge to combine a template and an optional field [26].

2.2 Retrieval-based

The retrieval-based Chatbots select the best matching answer for the user's question by searching a pre-constructed conversational repository [27]. Lowe et al. [28] developed a retrieval-based Chatbot using the Term Frequency
- Inverse Document Frequency (TF-IDF) method. The TF-IDF vectors of each candidate's question and answer are computed by concatenating all TF- IDF scores. The candidate answers with the highest cosine similarity to the question vector are selected as the final answers. Lu et al. [29] proposed an architecture to overcome the short- text matching problem of the developed models. Later, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN) and its extensions such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) were widely used in this field [23]. For instance, Zhou et al. proposed an approach by adding an attention mechanism to the deep network to provide question and answer matching [30].

2.3 Learning-based

This approach is usually based on Seq2Seq learning model. In the case of long sentences and conversations (more than 20 words), all essential information of a source sentence should be compressed into a fixed-length vector which is challenging [20]. To tackle this challenge, different approaches have been suggested by researchers. For instance, making the structural changes, such as adding the word embedding matrix [32], modification of the encoder or decoder [33, 34], and adding the attention mechanism [11, 13, 14, 35] are some of the solutions suggested by researchers.

2.4 Generative-based

Generative-based methods learn the answer distribution using the generative models, such as SeqGAN [21] and StepGAN [22]. These models use the Reinforcement Learning (RL) techniques. In the SeqGAN, The RL reward signal comes from the discriminator, which is judged on a complete sequence, and fed back to the

intermediate state action steps using Monte Carlo search. However, the Monte Carlo Tree Search (MCTS) used in SeqGAN has a high computational efficiency. In the StepGAN approaches, the Generative adversarial network (GAN) is evaluated in a step-wise manner. In this way, the discriminator is modified to automatically assign scores and determine the suitability of each generated subsequence. Wu and Wang added a new loss function (Truth guidance) to achieve a closer generated text to the real data [41].

3. METHODOLOGY

A vanilla GAN model consists of a generator (G) and a discriminator (D). Considering the scope of this paper, the GAN input is a question in the form of sequence x. The discriminator learns to maximize score (x, y^*) and minimize score

$D(x, y\hat{})$, while the generator learns to generate answer $y\hat{}$ to maximize $D(x, y\hat{})$ as expressed in Eq. (1):

$$minmax\,[log(x, y^*)] + E[\log(1 - D(x, y\hat{}))] \quad (1)$$

where (x, y^*) is the joint probability distribution of (x, y). $x \sim PR\ (x)$ denotes the probability distribution of x from training data. As the GAN models may never converge and have a problem of mode collapses, different variants of the GAN model have been suggested by researchers[48]. One of these variants is the Wasserstein GAN (WGAN), which is an extension of the GAN that seeks an alternate way of training the generator model to better approximate the distribution of data observed in a given training dataset. Instead of using a discriminator to classify or predict the probability of generated data as real or fake, WGAN changes or replaces the discriminator model with a critic that scores the reality or fakeness of a given data. The goal is to minimize the distance between the data distribution in the training dataset and the generated examples. This

method can promote stable training while working with gradients. Considering the superiority of the WGAN compared to the GAN model, we propose a novel architecture based on cWGAN and transformer model for generating answers in Chatbot. The intuition behind using WGAN in the proposed method is that there is no Sigmoid activation function to limit the values to 0 or 1 corresponding to real or fake. Also, the training process is more stable when a hybrid of WGAN with the transformer is used. As shown in Fig. 2, the proposed architecture consists of two modules: generator and discriminator which are connected in a single network. The reason for choosing WGAN in the proposed method and hybrid it with the transformer is to increase stability in the training process.

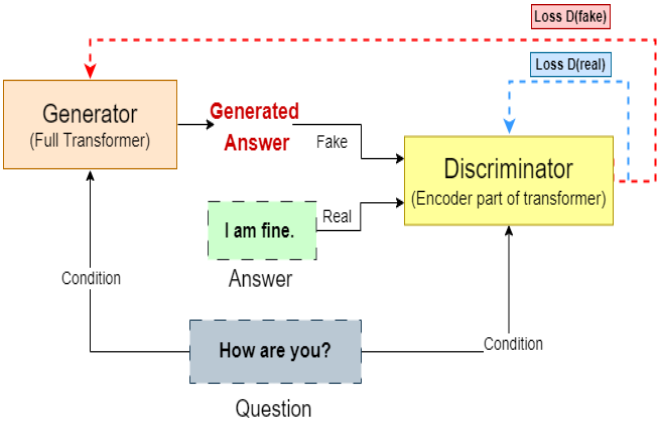

Fig. 1. General architecture of the proposed approach.

Generator modules

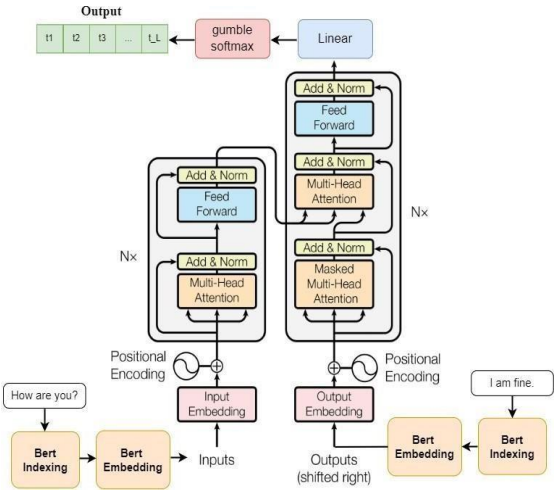

Fig. 2. Architecture of the proposed Generator in training phase.

Generator is a full transformer model that generates fake answers in test phase. The architecture of the Generator in training phase is illustrated in Fig. 3. The transformer architecture of this figure is adapted from the [20].

In the training phase, the encoder and decoder input embedding need to be prepared. Therefore, for the encoder input, the real questions are tokenized and embedded by the pretrained BERT model, including 12 layers, 768 features, and 12 heads. To increase the training speed, a linear model was adopted to reduce the dimensionality of the features to 64, the functionality of this linear model is shown in Eq. (2):

$$y = (\sum^{n} wixi\ +i{=}0\theta) \qquad (2)$$

Subsequently, the output of the linear model is concatenated with the position encoding, preserving the word's positions in the sentence. Positional encoding is a matrix, which gives context based on the word position in a sentence as Eq. (3) and Eq. (4):

$$(pos,2i) = \sin(pos/10000^{2i/d}model) \qquad (3)$$

$$(pos,2i{+}1) = \cos(pos/10000^{2i/d}model) \quad (4)$$

where "pos" refers to the position of the "word" in the sequence; while "d" means the size of the word/token embedding. "i" refers to each of dimensions of the embedding. "d" is fixed, while "pos" and "i" vary.

The same process with some variations is repeated for the decoder. In the decoder, instead of a question, we have a real answer in the input of the decoder. The transformer works slightly different during training and inference phase. During inference, only a question is presented as input sequence. There is not any real answer as target sequence that could be passed to the decoder. Since, decoder aims to generate an answer $y^{\wedge}$ as close as possible to the real answer, the output is generated in a loop and fed the output sequence from the previous timestep to the decoder in the next timestep till the end token.

In Generator, a full transformer with N = 8 identical layers and 16 heads is used. Generator module is first trained separately by Maximum Likelihood Estimation (MLE) to increase the convergence probability. Then, the model is fine- tuned by adversarial network to learn the answers

distribution. The linear transformation and Gumbel SoftMax function are utilized to convert the decoder output to the predicted next-token probabilities. The Gumbel-SoftMax distribution is a continuous distribution capable of approximating samples from a categorical distribution and providing a hard output by using an argmax function. So that, the output of decoder will be an index vector with L length, which is given as the input of the discriminator. In adversarial phase, the generator is updated upon obtaining the discriminator model. In GAN model, this is achieved by gradient likelihood ratios of objective function which can be derived by Eq. (5):

$$\square \quad = \nabla 1 \sum m \log[D(G(Q\ ))] \quad (5)$$

where LG shows the loss function of generator, m is the number of generated sequences while Q denotes the question regarded as condition data. As discussed before, we employ the WGAN, aiming to overcome the challenges of GAN model and increase stability. In this way, the objective function of WGAN

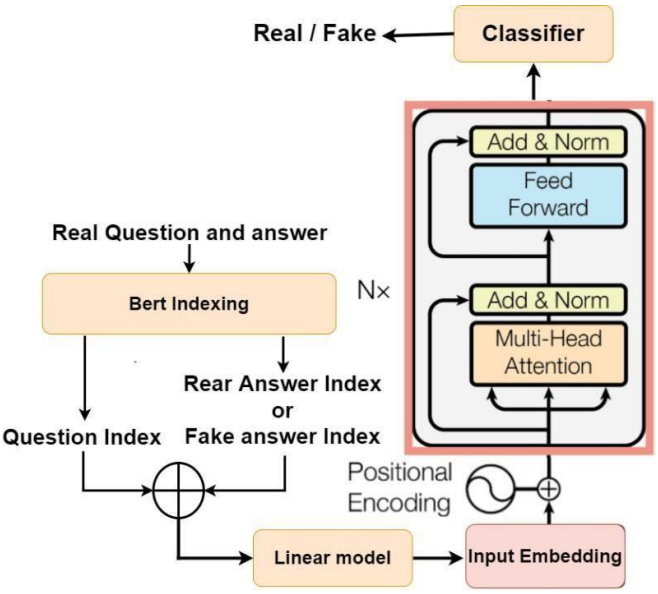

Fig. 3. Architecture of the Discriminator in the proposed model.

As we show in Fig. 4, in each iteration step, the discriminator is trained once with fake pair data and once with real pair data. Real question and answer pairs are tokenized and indexed by the pretrained BERT model. We cannot use BERT for embedding due to the computational complexity of the graph data. Therefore, the index matrix of fake

answer is concatenated with index matrix of question as fake pairs. Afterward, a linear model is adopted to reduce the dimensionality of the input matrix and sentence embedding. Then, the output of linear model is concatenated with position encoding matrix to serve as the input matrix for the first layer of the encoder. The output of the last layer of the encoder is fed to a classifier which is a linear network. This network has two scores outputs: the reality or fakeness of a given answer.

In the inference phase, we only have generator module while the discriminator model is removed. The decoder architecture of this Fig. 4 is adapted from the [20].

4. RESULT

To demonstrate the performance of the generative Chatbot model, this section presents the details of the dataset and the results in comparison with other methods. First, the implementation details are explained. Then, two datasets used for the evaluation, are briefly introduced, followed by the used evaluation metrics. Finally, the results of the proposed architecture are compared with the state- of-the-art models.

User input	Chatbot answer
hello Mr. parker how are you?	hello Ju thank you
I heard that was you good?	well, it was good seeing you
how much your goanna takes?	I do not know how much do you want
hey	hi what are you doing right now
how many girlfriends did you have?	I do not know truly
feeling better?	I actually not believe it

Table 1. Details of the parameters used in proposed architecture

Parameters	value	Parameters	value
Learning rate	0.00005	Number of layers	8
Batch size	64	Dataset split ration for test data	20%
Epoch numbers	400	Sentence Max Length	30
Processing way	GPU	Number of Heads	16
Dropout	0.5	BERT features size	768

Implementation details

Evaluations were carried out on a Core (TM) i5-12600K with 128GB RAM in Microsoft Windows 11 operating system and Python software with NVIDIA GeForce RTX 3090. The PyTorch library was used to implement the model. The implementation parameters are listed in Table 1.

4.1 Datasets

Two datasets, Cornell Movie Dialogs Corpus and Chit-Chat, are employed to evaluate the proposed model. Cornell Dataset contains a large metadata-rich collection of fictional conversations extracted from raw movie scripts. This dataset includes 220,579 conversational exchanges between 10,292 pairs of movie characters from 617 movies and a total of 304,713 utterances. The Chit-Chat dataset also encompasses 7,168 conversations from 258,145 utterances and 1,315 unique participants. This dataset was from the Chit-Chat Challenge of the BYU Perception, Control, and Cognition Laboratory.

4.2 Experimental results

For investigating the generated answers of the proposed Chatbot, various questions from different domains were asked from the proposed Chatbot. Table shows the results of the Chatbot trained using Cornell dataset in two areas of greeting and general questions and conversations. In addition, Table shows the results of the proposed Chatbot trained on the Chit-Chat dataset in two areas of greeting and general questions and conversation.

5. REFERENCE

1. Tran, A.D., J.I. Pallant, and L.W. Johnson, Exploring the impact of chatbots on consumer sentiment and expectations in retail. Journal of Retailing and Consumer Services, 2021. 63: p. 102718.
2. Miklosik, A., N. Evans, and A.M.A. Qureshi, The Use of Chatbots in Digital Business Transformation: A Systematic Literature Review. IEEE Access, 2021. 9: p. 106530-106539.
3. Okonkwo, C.W. and A. Ade- Ibijola, Chatbots applications in education: A systematic review. Computers and Education: Artificial Intelligence, 2021. 2: p. 100033.
4. Mogaji, E., et al., Emerging- market consumers' interactions with banking chatbots. Telematics and Informatics, 2021. 65: p. 101711.
5. Tsai, M.-H., et al., Four-Stage Framework for Implementing a Chatbot System in Disaster Emergency Operation Data Management: A Flood Disaster Management Case Study. KSCE Journal of Civil Engineering, 2020. 25.
6. Ayanouz, S., B.A. Abdelhakim, and M. Benhmed, A Smart Chatbot Architecture Based NLP and Machine Learning for Health Care Assistance. Niss2020.
7. Adamopoulou, E. and L. Moussiades, Chatbots: History, technology, and applications. Machine Learning with Applications, 2020. 2: p. 100006.
8. Ramesh, K., et al., A Survey of Design Techniques for Conversational Agents, in In Information, communication and computing technology. 2017: Singapore: Springer. p. 336–350.
9. Masche, J. and N.-T. Le. A Review of Technologies for Conversational Systems. in International Conference on Computer Science, Applied Mathematics and Applications. 2017. p. 2000–2004.
10. Dhyani, M. and R. Kumar, An intelligent Chatbot using deep learning with Bidirectional RNN and attention model. Materials Today: Proceedings, 2021. 34: p. 817-824.