



Design of a Modified Transformer Architecture Based on Relative Position Coding

Wenfeng Zheng¹ · Gu Gong¹ · Jiawei Tian¹ · Siyu Lu¹ · Ruiyang Wang¹ · Zhengtong Yin² · Xiaolu Li³ · Lirong Yin⁴

Received: 23 May 2023 / Accepted: 2 October 2023
© The Author(s) 2023

Abstract

Natural language processing (NLP) based on deep learning provides a positive performance for generative dialogue system, and the transformer model is a new boost in NLP after the advent of word vectors. In this paper, a Chinese generative dialogue system based on transformer is designed, which only uses a multi-layer transformer decoder to build the system and uses the design of an incomplete mask to realize one-way language generation. That is, questions can perceive context information in both directions, while reply sentences can only output one-way autoregressive. The above system improvements make the one-way generation of dialogue tasks more logical and reasonable, and the performance is better than the traditional dialogue system scheme. In consideration of the long-distance information weakness of absolute position coding, we put forward the improvement of relative position coding in theory, and verify it in subsequent experiments. In the transformer module, the calculation formula of self-attention is modified, and the relative position information is added to replace the absolute position coding of the position embedding layer. The performance of the modified model in BLEU, embedding average, grammatical and semantic coherence is ideal, to enhance long-distance attention.

Keywords Relative position embedding · Natural language processing · Attention mechanism

1 Introduction

Natural Language Processing (NLP) is a critical area of research that aims to enable machines to emulate human language and engage in seamless conversations with humans. This encompasses the capacity to read, comprehend, and fluently use language, master, and apply knowledge, and engage in logical thinking and inference [1–3]. Improved language intelligence through deep learning methods not only enhances a computer's ability to comprehend language

but also facilitates emotional expression and logical reasoning. Consequently, there are numerous potential applications for natural language processing solutions based on deep learning.

In the field of NLP, the adoption of recurrent neural networks (RNN) [4], attention mechanisms [5], and transformers [6] within end-to-end dialogue systems has significantly elevated the language comprehension and expression capabilities of these systems. In the earlier stages, RNN-based language models (RNNLM) gained prominence, achieving breakthroughs in NLP tasks. However, researchers soon encountered challenges related to long-range dependencies during model training. This arose from the tendency of weight parameters in RNN-based models to approach extremes, resulting in slow convergence and imprecise training outcomes. The introduction of long–short-term memory (LSTM) [7–10] addressed this issue. LSTM, a variant of RNN, is better suited for processing lengthy sequences due to its architectural design, which incorporates three gate structures (input gate, output gate, and forgetting gate) for controlling information flow.

The concept of sequence-to-sequence (Seq2Seq) [11, 12] models emerged in 2014 as a method to generate

✉ Wenfeng Zheng
winfirms@uestc.edu.cn

✉ Lirong Yin
lyin5@lsu.edu

¹ School of Automation, University of Electronic Science and Technology of China, Chengdu 610054, China

² College of Resource and Environment Engineering, Guizhou University, Guiyang 550025, China

³ School of Geographic Science, Southwest University, Chongqing 400715, China

⁴ Department of Geography and Anthropology, Louisiana State University, Baton Rouge, LA 70803, USA

sequences based on given input sequences. Initially applied to machine translation, Seq2Seq models addressed the challenge of handling variable-length input and output sequences. Over time, these models have shown promise in other NLP tasks, such as text summarization and dialogue generation. However, during the decoding phase, limitations were identified. The initial approach relied heavily on the last hidden layer state of the encoder, resulting in suboptimal information utilization [13]. In addition, when processing long input sequences, the fixed-length semantic vector struggled to retain critical feature information, leading to reduced accuracy. To overcome these issues, attention mechanisms were introduced [14].

The concept of attention mechanisms was initially proposed by Bahdanau et al. for machine translation and later improved by Luong et al. [15, 16]. Drawing inspiration from human selective attention, attention mechanisms mimic the human process of rapidly scanning and focusing on relevant information while disregarding irrelevant details. In the context of deep learning, attention mechanisms act as a resource allocation mechanism [17–19]. They dynamically redistribute the weight of information based on its importance, ensuring that critical information is given higher weight, while less important information is assigned lower weight. This feature extraction and sequential data analysis capability has found applications in various fields, including language modeling and image processing [20, 21].

Attention mechanism in the decoding process [22], each output not only depends on the fixed-size semantic vector encoded by the encoder, but also depends on the hidden layer state of the previous output unit and the corresponding hidden layer state of the current output unit in the decoding process. Attention is introduced into the Seq2Seq model to solve the problem that the original RNN often loses part of the input sequence information, and the accuracy of the model is improved. In the specific translation task [23, 24], the decoding phase is to translate one word by one word in the time series. When decoding one word, it will not have the same association with all the words in the source sequence. In the decoder phase, the selected reference contributes the most to the semantic vector of the current sequence word, rather than uniformly referring to all the semantic vectors.

The introduction of attention mechanisms into Seq2Seq models aimed to address limitations in retaining input sequence information and improve model accuracy. During the decoding phase, rather than uniformly considering all input semantic vectors, attention mechanisms enable the model to selectively focus on the most relevant reference for the current sequence word. Prior to this development, the most effective language models were based on Seq2Seq architecture with LSTM for modeling. However, this approach lacked parallel computing capabilities during

training, limiting the model's ability to meet the computational demands of increasingly larger corpora.

To fill the vacancy mentioned above. In this research, we introduced the implementation of a transformer-based generative dialogue system tailored for Chinese text. Theoretical foundations of the basic methods and process design were proposed. We designed a multi-turn generative dialogue system with an end-to-end structure that encodes natural language sentences into the model's vector space and generates sequences as output through the generative dialogue system's decoding process. When modeling and training the system, multi-turn statements were input in segments, and a self-regressive method was used to create a unidirectional generative language model. Generated words were continuously appended to the input until an end token was reached. We introduced a novel method to enhance long-distance attention within the dialogue system, replacing absolute position encoding in the position embedding module with relative position encoding. To test the effectiveness of relative position encoding in mitigating long-range information decay, we conducted experiments using multi-turn dialogue data sets, including the STC label data set and test data set. We compared the results with classical dialogue baseline models. The experimental results indicated that as the sequence length increased, accuracy improved, and the loss value decreased. This aligns with the expected outcomes of introducing relative position encoding, demonstrating that relative position encoding is better suited to handling long-text sequences compared to absolute position encoding. This underscores the effectiveness of our research optimization. In conclusion, the use of relative position encoding mitigates the issue of weak long-distance information, thereby enhancing the dialogue system's understanding of long-range information.

2 Related Work

2.1 End-to-end Dialogue Systems

Tomas et al. proposed a language model RNNLM based on RNN in 2010 [25, 26]. The model uses the vector of hidden states to record the historical information of word sequences. Hidden states can obtain long-range dependencies in the language. In the past, language models can only use the sliding window information of the front and back n words to predict the target words, while the advantage of the cyclic neural network is to fully use the context information to predict the target words. Sundermeyer et al. Introduced LSTM into the language model in 2012 and proposed LSTM-RNNLM [27]. The article mentions that LSTM has advantages over feed-forward neural networks, because it can utilize long-term contextual information. However, standard gradient

descent algorithms do not perform well in learning long-term dependencies due to the instability of gradient computation. To solve this problem, the article introduces an improved RNN architecture, namely, LSTM. LSTM controls the flow of information by introducing input gates, forget gates, and output gates, thereby avoiding the problems of gradient disappearance and gradient explosion.

2.2 Seq2Seq Encoder–Decoder Model

Seq2Seq is an encoder–decoder model [28]. The encoder and decoder are two cyclic neural networks, using the above-mentioned LSTM or its variant GRU. The recurrent neural network is an autoregressive network structure. The output of the last time in the sequence is the input of the next time. The function of the first recurrent neural network is to embed the input sequence into the fixed-length semantic vector space. The vector represents the characteristics of the input sequence. The network is named encoder. The other task of RNN is to generate the output sequence from the fixed-length vector. The network is named decoder. Seq2Seq model based on recurrent neural network (LSTM or GRU) has achieved good results.

In their research, Tianyu Liu et al. proposed a novel structure-aware seq2seq model for generating table-to-text descriptions [29]. This model improves generation performance by introducing an attention mechanism and a dual attention mechanism. The model can better utilize the structural information of the table and generate descriptions related to the content of the table. The results show that the model outperforms traditional statistical language models and basic seq2seq models in generative performance.

2.3 Transformer-Based Language Model

In 2017, the language model based on the transformer began to try not to rely on RNN and LSTM modeling [30]. Transformer was proposed by Google in its paper on machine translation tasks, and achieved very good translation results, which consists of a positionwise feed-forward network (FFN) layer and a multi-head attention layer. FFN is used in each position separately, which can guarantee the position information of each symbol in the input sequence during operation. The latter makes the model focus on information from different representation subspaces from different positions [31].

Transformer uses the self-attention mechanism to model the language model. Compared with RNN, self-attention mechanism not only increases the training parameters, but also realizes the parallelization through the complexity of space and parameters [32], which greatly accelerates the training efficiency of the model. In addition to being more parallelizable, the transformer establishes long-distance

dependence through the self-attention mechanism. Transformer model is unable to process long sequences due to its self-attention operation, which scales quadratically with the sequence length [33]. Relative position coding originated from Google's paper [34]. Shan et al. restricted the scope of self-attention to reduce the hybrid network model's consumption of memory and calculations and use the relative position encoding to improve robustness of the model. It is generally believed that relative position coding is inspired by absolute position coding [35]. Relative position information coding does not completely model the position information of each input but considers the relative distance between the current position and the position to be noticed when calculating attention, because natural language generally depends more on relative position. Therefore, relative position coding usually has excellent performance, and it is more flexible.

3 Methods

In this section, we will provide a detailed overview of the fundamental methods and process design employed in our study. The entire multi-turn generative dialogue system has been devised as an end-to-end structure. It takes natural language sentences, encodes them into a model vector space, and generates sequences as output through the generative dialogue system's decoding process. Furthermore, we propose the use of relative position encoding for self-attention computations, replacing the absolute position encoding in the dialogue system. This modification enhances long-range attention capabilities.

3.1 Dialogue System Implementation

First, a dialogue model network based on encoder–decoder is proposed, and the autoregressive model is adopted in the implementation process. In the autoregressive model, the statements in the dialogue system are defined as the following equation:

$$X = (x_1, x_2, \dots, x_t) \quad (1)$$

where X is a natural language sentence. x_i represents the word vector of the i th word, so the problem turns into encoding these sequences. Suppose the question is $X = (a, b, c, d, e, f)$, target output is $Y = (P, Q, R, S, T)$, the encoder–decoder structure of a basic dialogue system is shown in Fig. 1.

On the left of Fig. 1 is the encoder of the dialogue system, which is responsible for encoding the variable length input sequence as long as possible into a fixed-length semantic vector. Theoretically, this fixed-length vector should contain all the useful information of the input sentence. The decoder

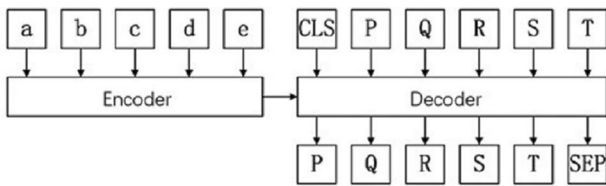


Fig. 1 Seq2Seq structure of dialogue system

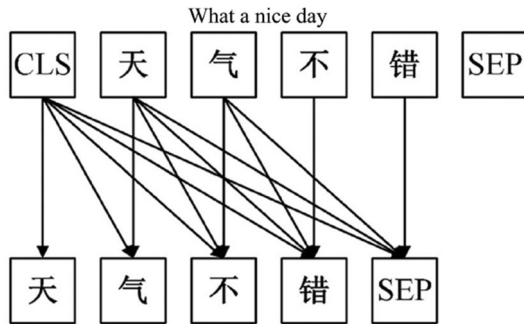


Fig. 2 Unidirectional language model

of the dialogue system is responsible for decoding the vector just encoded into our desired output sequence. Unlike the encoder, the decoder must be "unidirectional and recursive", because the decoding process is recursive. The specific process is as follows:

- 1) All output terminals start with the general identifier <CLS> tag and end with the <SEP> tag. These two identifiers are also regarded as one word;
- 2) Input <CLS> to the decoder, then obtain the hidden layer vector, mix this vector with the output of the decoder, and then input it to the interpreter. The result of the interpreter should be output as p;
- 3) Then input P into the decoder to obtain a new hidden layer vector, mix it with the output of the encoder again, and input it into the interpreter, which should output Q;
- 4) Recurse successively until the output of the interpreter is <SEP>.

In the decoding process of the decoder, the output language model is shown in the following equation:

$$p(P, Q, R, S, T) = p(P)p(Q|P)p(R|P, Q)p(S|P, Q, R)p(T|P, Q, R, S) \tag{2}$$

The so-called one-way language model, in a narrower sense, should be called positive language model. The crucial factor is that we cannot get "future" data. For example, in Eq. (2), there is no additional input during predicting P; When forecasting Q, you can only enter P; when forecasting R, you can enter P, Q; and so on.

As shown in Fig. 2, assume that the desired output result is $Y = (<CLS>, \text{天}, \text{气}, \text{不}, \text{错}, <SEP>)$, when

the decoder outputs, first the prediction result starts with the <CLS> identifier, input <CLS> into the decoder to get "气", continue to input into the encoder to get y in turn $Y = (\text{天}, \text{气}, \text{不}, \text{错}, <SEP>)$.

In the basic architecture, the transformer model is used to implement Seq2Seq. At this time, some key prior knowledge is introduced: considering that the input language and output language are Chinese, the hidden layer of encoder and decoder can share parameters and share the same set of word vectors, which will greatly reduce the number of parameters. The dialogue system is realized through multi-layer transformer decoder.

Considering that the dialogue system is suitable for multiple rounds of dialogue tasks, there is a context sentence segment with multiple rounds in the dialogue in the input text, which is expressed in English segment. The dialogue system introduces segment-level recurrence mechanism (SLRM), which stores the information of the previous segment every time and splices it with the information of the current segment. Suppose that there are two segments with length L in a sample data text, expressed as $s_\tau = (x_{\tau,1}, x_{\tau,2}, \dots, x_{\tau,L})$ and $s_{\tau+1} = (x_{\tau+1,1}, x_{\tau+1,2}, \dots, x_{\tau+1,L})$. Suppose the hidden layer information stored in s_τ is represented as $h_\tau \in R^{L \times D}$, D represents the dimension of the hidden layer vector, then the calculation method of $s_{\tau+1}$ is as Eqs. (3)–(5):

$$\tilde{h}_{\tau+1} = [SG(h_\tau) \circ h_{\tau+1}] \tag{3}$$

$$q_{\tau+1}, k_{\tau+1}, v_{\tau+1} = \tilde{h}_{\tau+1} W_q, \tilde{h}_{\tau+1} W_k, \tilde{h}_{\tau+1} W_v \tag{4}$$

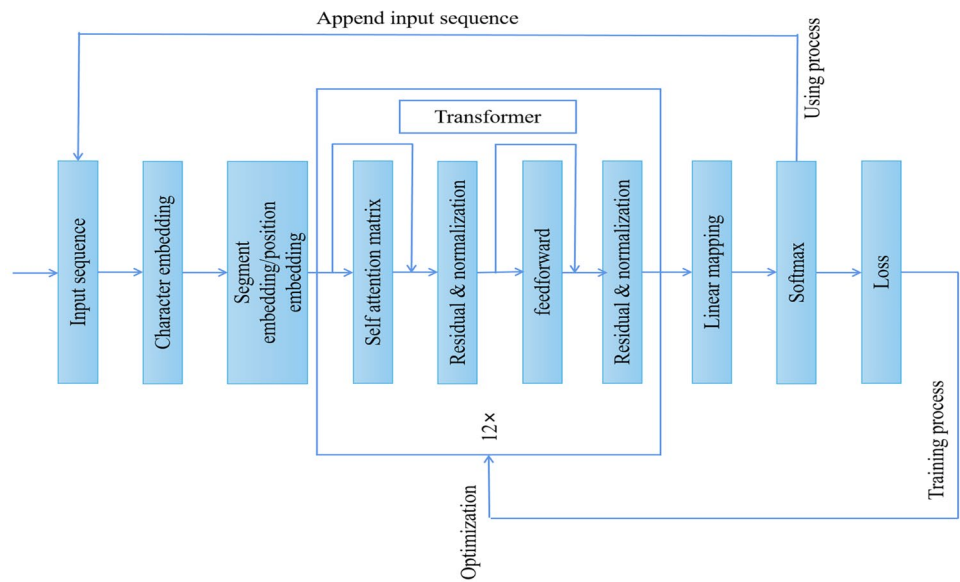
$$h_{\tau+1} = Transformer(q_{\tau+1}, k_{\tau+1}, v_{\tau+1}) \tag{5}$$

SG represents a stop gradient, which means that the parameters of the previous segment remain unchanged. The length of the hidden layer is increased through sequence splicing, and then the whole enters the transformer model for training. The specific process of transformer is shown in Fig. 3.

To sum up, the network structure of the dialogue system is mainly composed of the input layer, feature splicing layer, transformer decoder layer and output layer. The specific steps are:

- 1) Each data sample is spliced by multiple rounds of dialogue text. The LCCC (large-scale cleaned Chinese conversation) corpus data [36] preprocessing process is used to splice multiple rounds of dialogue into a language sequence for natural language processing, that is, first, take the [CLS] tag as the starting character, extract continuous dialogue sentences and fill in the input sample, insert the [SEP] tag between the sentences of different speakers, and set the maximum length. Note that the sequence length of the input sample is N.

Fig. 3 Flow chart of transformer-based generative dialogue system



2) To encode the data samples, first organize all words into a word table. By default, the word is the minimum granularity unit of the word vector. After the word table is established, record the number of word tables as V , and convert each word into a single hot coding vector to obtain $N \times V$ size matrix as a training sample. The specific operation is to set the value at the index i dimension to 1 and the others to 0. Taking Fig. 2 as an example, it is assumed that the processed sample is $Y = (\langle CLS \rangle, \text{天, 气, 不, 错}, \langle SEP \rangle)$, then $N = 6, V = 6$. The unique heat code in this scenario is described as the following equation:

$$[\langle CLS \rangle, \text{天, 气, 不, 错}, \langle SEP \rangle] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (6)$$

3) Learn the word embedding matrix W , and transform the unique hot coding into a word vector suitable for the subsequent transformer model by initializing it into a random word embedding matrix. The word embedding matrix $X_{WE} = XW$ of $N \times D$ is obtained by embedding the input words of the unique encoding into the network, where D represents the embedding dimension of the word embedding vector, and W is the word embedding matrix, with the size of $V \times D$.

4) Add segment embedding code. Segment embedding code indicates different roles of dialogue, which is represented by SegmentID. The specific vector content is a D -dimensional line vector filled with all 0 or all 1, where 0 or 1, respectively, represents the questioner or respondent, and N D -dimensional row vectors are spliced into a

segment embedding matrix X_{SE} of $D \times N$ according to the statement sequence.

5) To enhance the word vector representations with position coding information, a different approach is required when compared to cyclic neural networks. The transformer model, unlike cyclic neural networks, relies on the self-attention module and does not employ recursive operations. Consequently, it is unable to naturally capture timing information within the input text sequence and lacks inherent positional information for different word vectors in sentences. To address this limitation, the position information for each word needs to be incorporated into the word vectors. This enables the transformer model to distinguish the temporal sequence relationships among words within the current word order. To achieve this, position embedding is introduced, with the dimension of the position embedding set to $N \times D$. The method employed is the trigonometric function-based absolute position coding, often referred to as sinusoidal position coding [37, 38]. This technique integrates positional information into the input by performing a linear transformation using both sine and cosine functions, as shown in the following equation:

$$\begin{cases} p_{k,2i} = \sin\left(\frac{k}{10000^{\frac{2i}{d}}}\right) \\ p_{k,2i+1} = \cos\left(\frac{k}{10000^{\frac{2i}{d}}}\right) \end{cases} \quad (7)$$

Equation (7) is the absolute position information coding formula; k represents the position of the word in the sentence. The value range is $(0, N)$. d represents the dimension of the position vector, $p_{k,2i}, p_{k,2i+1}$ represents the $(2i)th, (2i + 1)th$ components of the position coding vector,

respectively, that is, the coding is calculated by sine function and cosine function in even dimension and odd dimension, respectively, to get X_{PE} . Thus, timing information with periodic changes is generated. The position is embedded in the dimension of the length n of the sample sequence. With the increase of the dimension number, the cycle will become slower and slower. Therefore, the model containing position texture information will be generated in even and odd dimensions. From this, the dependence between positions and the timing characteristics of natural language can be learned.

6) Feature fusion, calculate the matrix after adding location information and segmentation information, as shown in the following equation:

$$\begin{pmatrix} X_{WE} & X_{SE} & X_{PE} \end{pmatrix} \begin{pmatrix} W_E \\ W_S \\ W_P \end{pmatrix} = X_{WE}W_E + X_{SE}W_S + X_{PE}W_P \quad (8)$$

To avoid data loss, three information matrices can be spliced together $(X_{WE} \ X_{SE} \ X_{PE})$, due to formula (7), the three information matrices are added directly, as shown in the following equation:

$$X_E = X_{WE} + X_{SE} + X_{PE} \quad (9)$$

7) Input X_E into language expression layer is stacked by several layers of transformer decoder units, and the specific calculation of each layer module is as Eqs. (10)–(12):

Introduce the self-attention mechanism to calculate the attention moment matrix Z . First, X_E is multiplied by three $D \times D$ size weight matrices W^Q, W^K, W^V to obtain the query matrix Q , key matrix K and value matrix V :

$$Q = X_E W^Q = [q_1, q_2, \dots, q_N] \quad (10)$$

$$K = X_E W^K = [k_1, k_2, \dots, k_N] \quad (11)$$

$$V = X_E W^V = [v_1, v_2, \dots, v_N] \quad (12)$$

The specific formula of attention mechanism is as the following equation:

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (13)$$

Suppose a sentence is $X = (x_1, x_2, x_3, x_4)$, that is, there are four-word vectors in the statement. After the operation from Eqs. (10) to (12), the respective query vector Q , key vector K and value vector V are obtained, respectively. As shown in Fig. 4:

When calculating the self-attention vector of the first word x_1 , it is necessary to calculate the dot product between the key vector of all words and the query vector of the current

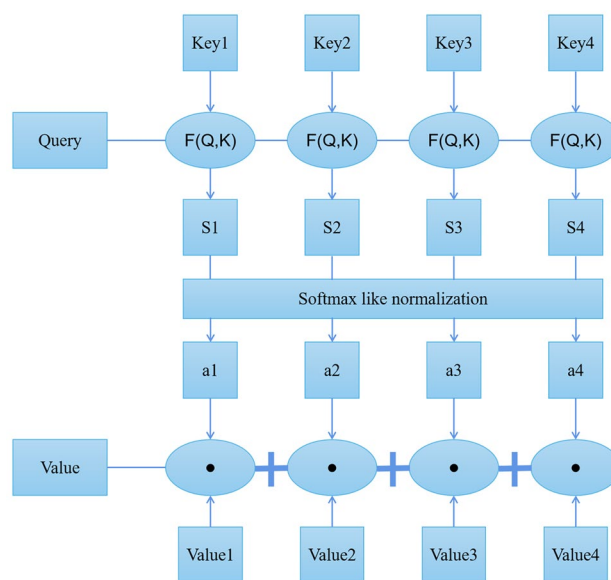


Fig. 4 Specific process of attention calculation

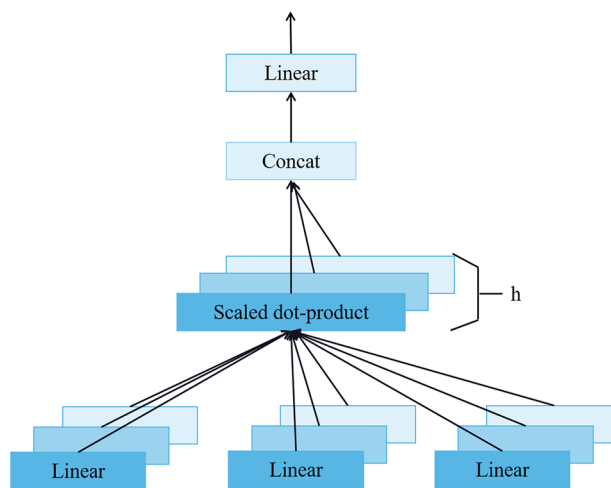


Fig. 5 Multi-head attention

word x_1 to get the score. Each score is divided by the square root of $\sqrt{d_k}$ to get $S_i, i = 1, 2, 3, 4$. Then, calculate Softmax to normalize the scores of all words to $a_i, i = 1, 2, 3, 4$.

This study adds a multi-head attention matrix to the Q, K , and V matrices to improve the attention unit's ability to extract multiple semantics of a word [39–43]. The schematic diagram of multi-head attention is shown in Fig. 5. In this example, the implementation process of multi-head attention mechanism is to define the super parameter $h=3$ to represent the number of heads, divide D into h parts, and divide Q, K and V into parts $(Q_i, K_i, V_i), i = 1, 2, 3$ through linear mapping, calculate attention for each part. The process is shown in Eqs. (14)–(16):

$$Q_i = QW_i^Q, K_i = KW_i^K, V_i = VW_i^V, i = 1, \dots, h \tag{14}$$

$$head_i = Attention(Q_i, K_i, V_i), i = 1, \dots, h \tag{15}$$

$$Z = MultiHead(Q, K, V) = Concat(head_1, head_2, \dots, head_h)W_0 \tag{16}$$

W_0 is the weight of the linear layer, the h attention features are spliced and linearly projected to obtain the attention feature matrix Z .

Then connect the residuals, and the specific implementation is to add Z with X_E get the attention matrix and get $X_A = Z + X_E$. At the same time, layer normalization is performed. The function of standard normalization is to treat the hidden layer in the network as standard normal distribution and speed up the convergence of loss function in the training process. Obtain X'_A , as the following equation:

$$LayerNorm(x) = \alpha \times \frac{x_{ij} - u_i}{\sqrt{\sigma_i^2 + \epsilon}} + \beta \tag{17}$$

Parameters u_i and σ_i , respectively, represent the mean and standard deviation of each element x_{ij} , ϵ , is a minimum constant to prevent numerical calculation problems caused by division by 0, and α and β are trainable parameters to compensate for information loss caused by normalized.

Transfer the residual and normalized matrix X'_A to the feed-forward layer. The feed-forward module is a multi-layer perceptron (MLP), which has a hidden layer. The hidden layer matrix is obtained by two-layer linear mapping and

activation with the activation function ReLU, as shown in the following equation:

$$X_H = ReLU(Linear(Linear(X'_A))) \tag{18}$$

For matrix X_H is then connected with the residuals and added with X'_A to obtain $X'_H = X'_A + X_H$. The X'_H matrix is normalized and a new embedded matrix X_E is output.

After the multi-layer transformer module, the matrix X_{TE} of $D \times N$ is output. The processing steps of the multi-layer transformer module are summarized as follows: first, it is processed through the self-attention layer, and then transferred to the neural network layer. After the current transformer module is processed, it then transfers the vector to the next transformer module.

8) In the output layer, when the last transformer module generates the output, the model multiplies the output vector by the word embedding matrix W . Each row of the word embedding matrix corresponds to the word embedding vector in the model word table, and the attention score corresponding to each word in the word table is obtained by multiplication [44–46].

Finally, Softmax is used to predict word in the output dictionary, and the model uses cross entropy to update the parameters. In this way, the model completes a round and outputs a word. The model then continues to recurse until a complete sequence is generated, the upper limit n of the sequence is reached, or the terminator $\langle SEP \rangle$ is generated. Finally, the complete basic structure of the system is shown in Fig. 6. Table 1 shows the specific network parameters of the transformer-based generative dialogue system.

Fig. 6 Structure diagram of generative dialogue system based on transformer

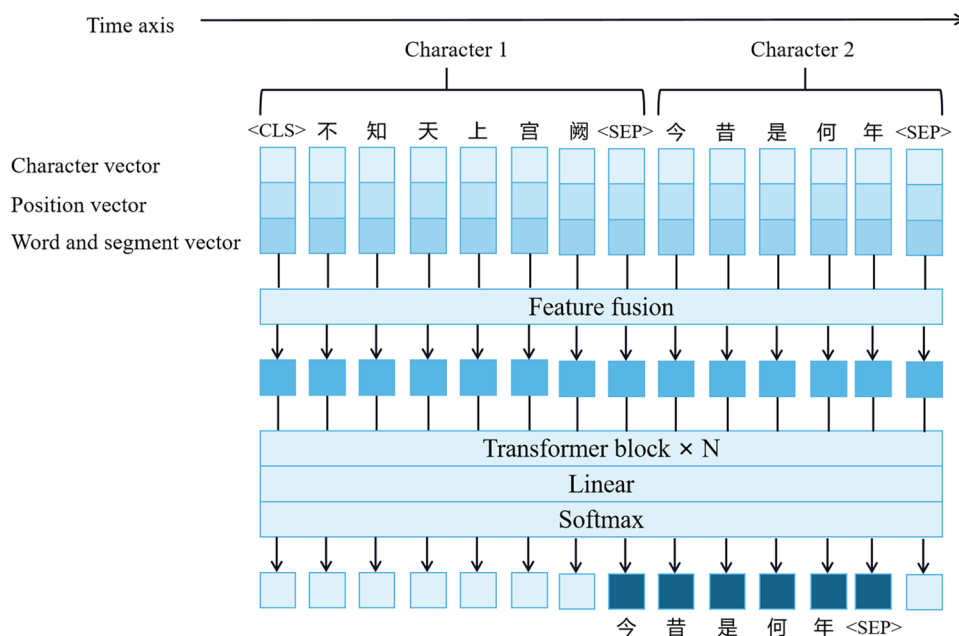


Table 1 Network structure of dialog model based on transformer

Order	Layer	Dimension	
(1)	Embeddings	$V = 13,088$	$D = 384$
(2)	Positional Encoding	256	384
(3)	Segment Encoding	256	384
(4)	Attn/c_attn	256	384×3
(5)	Attn/c_proj	256	384
(6)	Add&Norm	256	384
(7)	Mlp/c_fc	768	384×4
(8)	Mlp/c_proj	384×4	384
(9)	Add&Norm	256	384

As mentioned above, the transformer architecture can learn global information. The key lies in the self-attention mechanism. The self-attention mechanism calculates the encoded input sequence with each other to obtain the cosine similarity and form a similarity matrix of size n^2 . which represents the length of the input sentence sequence $[X, X]$. Compared with the spatial complexity $O(n)$ of RNN, the spatial complexity of the self-attention matrix is $O(n^2)$. The complexity of space and parameters realizes parallelism, and the increase of parameters of the self-attention matrix can contain more statement information, rather than the limitation of the result caused by the fixed-length semantic vector.

Each column of the attention matrix represents input and each row represents output. The matrix represents the correlation between input and output. Suppose you input "你想吃啥" and the reply is "白切鸡", then the above statements are spliced into $(\langle \text{CLS} \rangle, \text{你}, \text{想}, \text{吃}, \text{啥}, \langle \text{SEP} \rangle, \text{白}, \text{切}, \text{鸡}, \langle \text{SEP} \rangle)$

by language sequence preprocessing. When training unidirectional generative language model, input $(\langle \text{CLS} \rangle, \text{你}, \text{想}, \text{吃}, \text{啥}, \langle \text{SEP} \rangle)$, and predict in turn

(白, 切, 鸡) until $\langle \text{SEP} \rangle$ appears. Considering that current input cannot take advantage of "future" information. To generate a unidirectional language sequence, the input and output are staggered by one bit, as shown in Fig. 7a:

The white square represents 0. The first line indicates that "你" can only be related to the starting mark $\langle \text{CLS} \rangle$, the second line indicates that "想" is related to the starting mark $\langle \text{CLS} \rangle$ and "你", and so on.

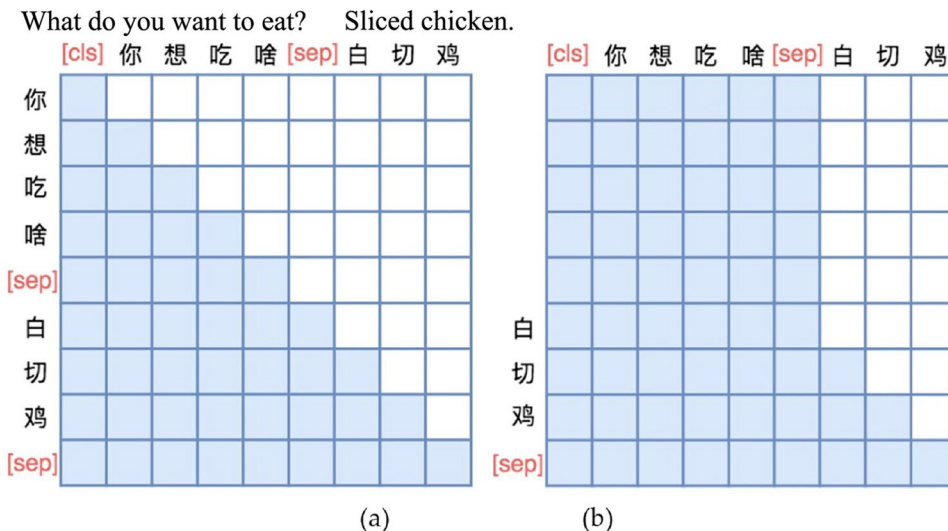
But the above model will also add the input $(\langle \text{CLS} \rangle, \text{你}, \text{想}, \text{吃}, \text{啥}, \langle \text{SEP} \rangle)$ to the prediction range, which belongs to additional constraints. The only thing that really needs prediction is (白, 切, 鸡). Therefore, this study refers to the idea of Mask for UNILM [23]. Design incomplete mask, only mask (白, 切, 鸡) part, reserve information of $(\langle \text{CLS} \rangle, \text{你}, \text{想}, \text{吃}, \text{啥}, \langle \text{SEP} \rangle)$ part. As shown in Fig. 7b:

The attention of the input part obtains two-way context information, while the output part is one-way attention. Therefore, in the process of complementing the length of blank remaining sentences and mask in the maximum sentence length, 0 is generally filled in. This process is called padding. However, there will be problems during Softmax. The reason is $e_0 = 1$ is an effective number. In this way, the padded part of Softmax participates in the operation, which is equivalent to allowing the invalid part to participate in the operation, which has an impact on Softmax calculation and results in deviation. Currently, it is necessary to cover up these invalid parts and do not participate in the calculation. The specific method is to add a large negative offset to the invalid part, as shown in Eqs. (19)–(21):

$$z_{\text{illegal}} = z_{\text{illegal}} + \text{bias}_{\text{illegal}} \tag{19}$$

$$\text{bias}_{\text{illegal}} \rightarrow -\infty \tag{20}$$

Fig. 7 Mask design. **a** Mask of one-way language model; **b** incomplete mask of this dialog system



$$e^{z_{illegal}} \rightarrow 0 \quad (21)$$

After the calculation of Eqs. (19)–(21), the masking part will not be affected after participating in Softmax calculation, and the calculation increment is still 0, to avoid the influence of invalid area on training accuracy.

3.2 Absolute Position Information Coding

The transformer-based attention model is the dot product of the similarity between vectors in the matrix, there is no timing information added by the recursive process of RNN. In most cases, the previous solution is to integrate the location information into the input, which constitutes the general practice of absolute location coding. However, when dealing with multiple rounds of QA dialogue, the dialogue length should be unlimited in theory, but the design of absolute location coding limits the length of the dialogue text, and the above memory effect is not ideal in the long text.

The following analyzes several commonly used absolute position codes.

1) One of the most concise schemes of absolute position coding is to directly train the position coding as a trainable parameter without designing the position coding formula. If the maximum length of the vector is N and the coding dimension is D , then initialize an $N \times D$ matrix as position vector to update with the training process. The current BERT, GPT and other models use this kind of coding. The earliest Facebook paper in 2017 used this method [47].

However, for this training absolute position coding, its disadvantage is that it has no scalability. If the maximum length formula of pre training is set to 512, it can only process sentences with a length of 512 at most. There is no matching location information for locations longer than 512. The solution is to randomly initialize the position code of more than 512, and then conduct training fine-tuning.

2) Trigonometric function is another scheme of position coding, also known as sinusoidal position coding [30]. The coding is calculated by sine function and cosine function in even and odd dimensions, respectively, to generate timing information with periodic changes. The position is embedded in the dimension of length N of the sample sequence. With the increase of the dimension number, the cycle will become slower and slower. Therefore, the model containing position texture information is generated in even and odd dimensions, as shown in Fig. 8. Thus, the dependency between positions and the timing characteristics of natural language can be learned.

It can be seen from Fig. 8 that the position coding of trigonometric function is characterized by the periodic generation law according to the time sequence. However, with the increase of dimension serial number, the periodic change

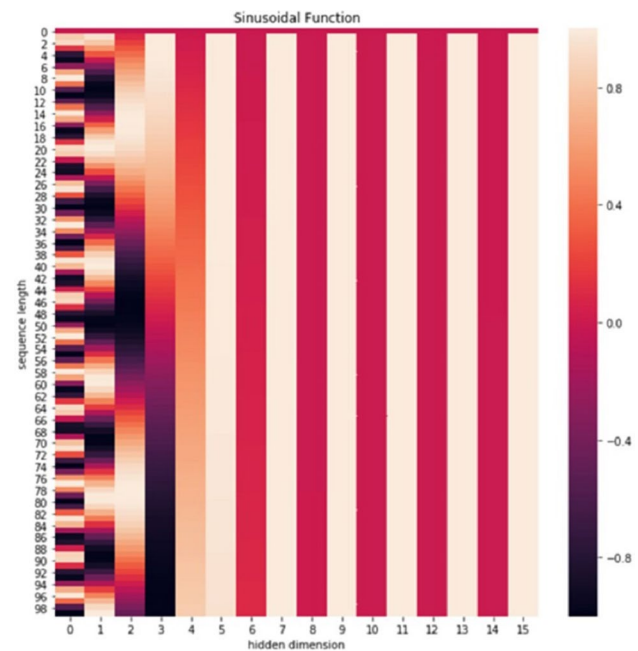


Fig. 8 Schematic diagram of timing information

will be slower and slower, which leads to the decline of the discrimination of position information for the input of long text in multiple rounds of dialogue.

3) Theoretically, the reason why the RNN model does not need location coding is that it naturally has the possibility of learning location information. Therefore, assuming that a layer of RNN is added before the input word vector enters the model, and then input into the transformer module, the location information can be obtained theoretically, and the location coding is no longer needed. Similarly, RNN model training can be used to learn absolute position coding. ICML2020's paper [48] continues to develop this idea and proposes to model the position coding by means of differential equation (ODE). This method is called FLOATER. FLOATER belongs to recursive call model, so this differential equation is also called neural differential equation. In terms of basic theory, recursive location coding also has better scalability, and it also has better flexibility than trigonometric location coding. Obviously, recursive location coding sacrifices a certain degree of parallelism and will bring a speed bottleneck in theory.

In addition, language model performance improvement techniques other than positional encoding continue to develop. Hassan I. Abdalla et al. [49, 50] proposed a scheme to improve text classification neural network using BoW, and improve the performance of text recognition and matching by integrating similarity measures with machine learning models. Internal evaluation of the ensemble model against the baseline model demonstrates

that the above method has good optimization performance for neural networks for NLP.

3.3 Design of Relative Position Information Coding

In this section, we will solve the problem of location information by introducing the design of relative location coding representation and detailing this process. A new attention calculation method of relative position coding is introduced to replace the absolute position coding of position embedding layer.

The disadvantage of absolute position coding is that it will produce remote attenuation. The larger the relative distance, the weaker the correlation between the inputs. The reason why periodic trigonometric functions appear and show attenuation trend is that the integral from high-frequency oscillation asymptotically approaches 0.

The assumed model is $f(\dots, x_m, \dots, x_n, \dots)$, where, x_m, x_n represent the m th and n th input words, respectively. Now, we discuss the generality, and set f as a scalar function for calculation. This study uses transformer-based attention mechanism, so the function f has the characteristics of total symmetry, that is, for any m and n , there are shown in the following equation:

$$f(\dots, x_m, \dots, x_n, \dots) = f(\dots, x_n, \dots, x_m, \dots) \tag{22}$$

Full symmetry is the main reason why transformer cannot recognize the position. Specifically, the function naturally satisfies the identity $f(x, y) = f(y, x)$, so that it is impossible to distinguish whether the input is (x, y) or (y, x) from the result.

Therefore, to break this symmetry, it is necessary to add a position coding information. One feasible scheme is to add a position determined coding vector to each position, as shown in the following equation:

$$\tilde{f}(\dots, x_m, \dots, x_n, \dots) = f(\dots, x_m + p_m, \dots, x_n + p_n, \dots) \tag{23}$$

In general, assuming that all position encoding vectors are not the same, full symmetry does not hold. This means that we can use \tilde{f} instead of f to address the input from positional timing information. To simplify the problem, we only consider the position encoding at two positions, m and n , and introduce it as a perturbation term, expanding it to the second-order Taylor term, as shown in the following equation:

$$\tilde{f} \approx f + p_m^\top \frac{\partial f}{\partial x_m} + p_n^\top \frac{\partial f}{\partial x_n} + \frac{1}{2} p_m^\top \frac{\partial^2 f}{\partial x_m^2} + \frac{1}{2} p_n^\top \frac{\partial^2 f}{\partial x_n^2} + p_m^\top \frac{\partial^2 f}{\partial x_m \partial x_n} p_n \tag{24}$$

From Eq. (24), item 1 is independent of location, items 2 to 5 only depend on a single location, so they

only depend on absolute location information, and item 6 owned the theinteractionitemof p_m, p_n at the same time, record it as $p_m^\top p_n$, it will be analyzed later, and it is expected to express certain relative position information on this basis. Suppose I is the identity matrix, at this time $p_m^\top p_n^\top = p_m^\top p_n = \langle p_m, p_n \rangle$ is the inner product of two position codes. It is hoped that in this simple example, this item represents the relative position information, that is, there is a function g as the following equation:

$$\langle p_m, p_n \rangle = g(m - n) \tag{25}$$

Here, p_m, p_n is a d -dimensional vector, assuming $d = 2$, then for a two-dimensional vector, it is derived with the help of the complex number, that is, the vector $[x, y]$ is regarded as the complex number $x + yi$. According to the algorithm of complex number multiplication, get the following equation:

$$\langle p_m, p_n \rangle = Re[p_m p_n^*] \tag{26}$$

p_n^* is the conjugate complex of p_n . Re represents the real part of the complex number. To satisfy Eq. (26), it can be assumed that there is a complex number q_{m-n} , as shown in the following equation:

$$p_m p_n^* = q_{m-n} \tag{27}$$

In this way, taking the real part on both sides and we can obtain the following equation:

$$r_m r_n e^{i(\phi_m - \phi_n)} = R_{m-n} e^{i\Phi_{m-n}} \tag{28}$$

To solve this equation, the exponential form of the complex number can be used. Suppose $p_m = r_m e^{i\phi_m}$, $p_n^* = r_n e^{-i\phi_n}$, $q_{m-n} = R_{m-n} e^{i\Phi_{m-n}}$, then we can get the following equation:

$$\begin{cases} r_m r_n = R_{m-n} \\ \phi_m - \phi_n = \Phi_{m-n} \end{cases} \tag{29}$$

For Eq. (28), substitute $n = m$ to get $r_m^2 = R_0$, r_m is a constant, set to 1 for simplicity; for the second equation, let $n = 0$, then $\phi_m - \phi_0 = \Phi_m$, if $\phi_0 = 0$, then $\phi_m = \Phi_m$, which is the same as $\phi_m - \phi_n = \phi_{m-n}$. Substitute $n = m - 1$, then $\phi_m - \phi_{m-1} = \phi_1$. So that $\{\phi_m\}$ is an arithmetic sequence. Therefore, the solution of position coding in two-dimensional case is shown in the following equation:

$$p_m = e^{im\theta} \Leftrightarrow p_m = \begin{pmatrix} \cos m\theta \\ \sin m\theta \end{pmatrix} \tag{30}$$

Because the inner product has the characteristic of linear superposition, the higher dimensional even dimensional position coding can be expressed as a combination of multiple two-dimensional position codes to obtain formula (39).

Equation (31) choose $\theta_i = 10000^{-2i/d}$, this form has a good property: it makes $\langle p_m, p_n \rangle$ tends to 0 when $|m-n|$

gets larger. The larger the relative distance, the weaker the correlation. The reason is that the high-frequency oscillation integral gradually tends to 0, i.e., Eq. (32):

$$p_m = \begin{pmatrix} e^{im\theta_0} \\ e^{im\theta_1} \\ \dots \\ im\theta \\ d/2 \\ e \quad -1 \end{pmatrix} \Leftrightarrow p_m = \begin{pmatrix} \cos m\theta_0 \\ \sin m\theta_0 \\ \cos m\theta_1 \\ \sin m\theta_1 \\ \dots \\ \cos m\theta_{d/2-1} \\ \sin m\theta_{d/2-1} \end{pmatrix} \tag{31}$$

$$\begin{aligned} \langle p_m, p_n \rangle &= \text{Re} [e^{i(m-n)\theta_0} + e^{i(m-n)\theta_1} + \dots + e^{i(m-n)\theta_{d/2-1}}] \\ &= \frac{d}{2} \text{Re} \left[\sum_{i=0}^{d/2-1} e^{i(m-n)10000^{-2i/d}} \frac{1}{d/2} \right] \\ &\sim \frac{d}{2} \text{Re} \left[\int_0^1 e^{i(m-n)10000^{-t}} dt \right] \end{aligned} \tag{32}$$

The general attention with absolute position coding is as Eqs. (33)–(35):

$$\begin{cases} q_i = (x_i + p_i)W_Q \\ k_i = (x_i + p_i)W_K \\ v_i = (x_i + p_i)W_V \end{cases} \tag{33}$$

$$a_{i,j} = \text{softmax}(q_i k_j^T) \tag{34}$$

$$O_i = \sum_j a_{i,j} v_j \tag{35}$$

SoftMax is used to normalize the row dimension j , x_i , p_i are line vectors, p_i indicates the added location information. Preliminary expand $q_i k_j^T$, the expansion is shown in the following equation:

$$q_i k_j^T = (x_i + p_i)W_Q W_K^T (x_i + p_i)^T = (x_i W_Q + p_i W_Q) (W_K^T x_j^T + W_K^T p_j^T) \tag{36}$$

To introduce relative position information, the structure of Eq. (36) is modified to the following equation:

$$a_{i,j} = \text{softmax}(x_i W_Q (x_j W_K + R_{i,j}^K)^T) \tag{37}$$

That is, remove the position of the first item and the position information of second item $p_i W_K$ is changed to binary position vector $R_{i,j}^K$. Then, expand the following equation:

$$O_i = \sum_j a_{i,j} v_j = \sum_j a_{i,j} (x_j W_V + p_j W_V) \tag{38}$$

Change $p_i W_V$ to $R_{i,j}^V$, get the following equation:

$$O_i = \sum_j a_{i,j} (x_j W_V + R_{i,j}^V) \tag{39}$$

As can be seen from the formula above, the so-called relative position is to change the vector that originally depends on binary coordinates (i, j) to only depend on the relative position distance $i - j$, and usually needs to be truncated to adapt to different arbitrary distances. Therefore, the expression of $R_{i,j}^K$ is shown in Eqs. (40) and (41):

$$R_{i,j}^K = p_K [\text{clip}(i - j, p_{min}, p_{max})] \tag{40}$$

$$R_{i,j}^V = p_V [\text{clip}(i - j, p_{min}, p_{max})] \tag{41}$$

Through the above modification, although only a limited number of position coding information are obtained, the relative position of any length can be expressed, p_K, p_V is the relative position code of trigonometric function formula. The specific definition of relative position code is shown in Eqs. (42) and (43):

$$p_{ij}[2k] = \sin\left(\frac{j - i}{10000 \frac{2k}{dc}}\right) \tag{42}$$

$$p_{ij}[2k + 1] = \cos\left(\frac{j - i}{10000 \frac{2k}{dc}}\right) \tag{43}$$

4 Experiments and Results

The purpose of this experiment is to test the role of relative position coding in long-distance multi-round conversation scenarios, and to verify whether relative coding is better than absolute position coding in slowing down the decline of long-distance information.

4.1 Data Set and Environment

The pre-training of all models in this experiment used the LCCC corpus [36] as the training data set. In this study, the Chinese dialogue data set STC (short text conversation) [51] is selected for evaluation experiments, and the Chinese generative dialogue system and some classical dialogue baseline models are compared.

The short text dialogue corpus STC published by 2015 Huawei Noah's Ark laboratory is required to predict the reply under a given number of rounds of contextual dialogue

corpus. The data set contains 4.43 million Chinese dialogues. Each post has an average of 20 different replies, and there is a semantic gap between each reply, which is the main difference from the traditional parallel data set.

This section mainly uses the division method in STC and selects its test data as the test data set for this study. Compared with the labeled data set, the test data are characterized by sparse roles. As there is no labeled data set, the whole data set is manually labeled, and the tagger will actively guide the topic to talk about the characterization information. Therefore, most of the responses in the data set are characterization-related. However, in a real conversation scenario, it often does not involve a lot of characterization information in every chat. From this point of view, the test data set is closer to the characterization conversation in a real man-machine conversation.

The experimental environment used for the experiment are some open-source frameworks, including TensorFlow 2.1.0, keras2.3.1, bert4Keras0.9.8. Specifically, the experimental operation environment is shown in Table 2.

4.2 Experimental Evaluation Criteria

In this study, we first use the accuracy of generated tags (ACC) to evaluate the effectiveness of relative position coding, and then use objective and subjective evaluation methods for the final model.

In terms of objective evaluation indicators, we apply several commonly used dialogue evaluation methods, including PPL (perplexity score) [52], BLEU [53], Greedy Matching [54–56] and Embedded Average [57]. Some experiments show that the evaluation methods based on word embedding have higher correlations with human [58].

In the manual evaluation index [59], Grammatical and Semantic Continuity, Context Relevance and amount of information are used. 200 replies were sampled for each model, and 2 marked students were invited to manually evaluate these replies.

4.3 Performance of Chinese Dialogue System Based on Transformer

In this experiment, the model features a 12-layer transformer decoder architecture, with individual characters as

the smallest unit for word embeddings. It employs 12 multi-head attention heads, a vocabulary size of 13,088, character vector dimensions set at 384, and a maximum context length of 256 characters. The batch size used is 16. For training optimization, the dialogue system utilizes the Adam optimizer with a primary learning rate of 2×10^{-5} . The training is conducted over 100 epochs on the LCCC-base data set.

To validate the effectiveness of the proposed generative dialogue system in this study, three methods are introduced for comparison with the baseline model: Attn-Seq2Seq, transformer, and GPT. The baseline model is described as follows:

Attn-Seq2Seq [36]: This model is based on the traditional Seq2Seq architecture with the addition of an attention mechanism. It also employs a multi-turn dialogue approach to concatenate multiple segments, similar to the dialogue system in this chapter. It uses the <CLS> and <SEP> identifiers to segment the segments, and these specific identifiers have corresponding word embeddings. The concatenated history dialogue is encoded and decoded using a bidirectional LSTM as the basic unit for Seq2Seq.

Transformer [60]: The transformer model serves as the foundational architecture. This model has found broad applications in machine translation and dialogue generation. To ensure fairness in training the transformer model, a 12-layer transformer is used and trained for 100 epochs on the LCCC-base data set.

GPT-chatbot [61]: This model is based on the GPT2 architecture for generative dialogue. Each training data are "sequentially concatenated" following the approach of Microsoft's DialoGPT. The concatenated data are then input into the network for training. The model consists of 12 layers of GPT and is trained for 100 epochs on the LCCC-base data set.

All transformer-based models share the same parameter settings for the encoder and decoder structures. They are essentially like the GPT model parameters. The vocabulary size is 13,088, word vector dimensions are 384, the maximum context length is 256, and the batch size is 16.

By conducting experiments, the objective evaluation index experimental results as shown in Table 3 are obtained:

Table 3 displays the results of different language models in terms of perplexity (PPL), BLEU-2, BLEU-4, Dist-1, and Dist-2. A lower perplexity indicates smoother sentence

Table 2 Experimental environment

Processor	Intel(R) Core(TM) i7-9800X CPU @ 3.80 GHz
Memory	64 GB
Graphics card	NVIDIA GeForce GTX2080Ti
Operating system	Ubuntu 18.04.3 LTS
Development environment	Pycharm + Anaconda
Open source framework	TensorFlow2.1.0、Keras2.3.1、BERT4Keras0.9.8

Table 3 Objective evaluation index experimental results

Model	PPL	BLEU-2	BLEU-4	Dist-1	Dist-2
Attn-Seq2Seq	37.23	4.51	0.94	8.5	11.94
Transformer	22.30	6.72	3.14	8.8	12.11
GPT-chatbot	20.52	5.69	2.78	8.1	11.73
Ours	19.83	6.63	3.20	9.2	12.68

Table 4 Subjective evaluation results

Model	Syntax-semantic coherence	Contextual relevance	Information amount
Attn-Seq2Seq	1.13	0.92	1.17
Transformer	1.34	1.15	1.39
GPT-chatbot	1.59	1.20	1.40
Ours	1.64	1.33	1.42

Table 5 Experimental model setup

	Baseline	Char-word	Model of this study
Token unit	Chinese character	Word-character fusion	Chinese character
Location coding	Absolute position	Absolute position	Relative position

generation and better relevance to the topic. The best PPL is achieved by the dialogue model in this chapter, which, although only slightly outperforms the GPT2-chatbot, still holds an advantage. This can be attributed to the effectiveness of adding partial masking. In terms of perplexity performance, it surpasses other language models to some extent.

In BLEU-2 and BLEU-4 evaluations, this language model performs best in BLEU-4 but falls short of transformer in BLEU-2. This difference can be explained by the fact that BLEU was originally developed as an evaluation metric for translation, and it tends to favor shorter translation results. It may not handle morphologically rich sentences well, making it less friendly for generative dialogue models. Finally, in terms of diversity as measured by Dist-1 and Dist-2, this experimental model outperforms all baseline models.

In the human evaluation metrics, 200 responses were sampled for each model, and two annotators were invited to conduct manual evaluations on these responses. The results of the subjective evaluation metrics are presented in Table 4 as follows:

The experimental results from the manual subjective evaluations demonstrate that the dialogue system designed in this study outperforms the baseline models across all three metrics. The dialogue system in this chapter is capable of generating high-quality dialogue responses. It not only produces responses with sufficient information but also

maintains good fluency in the sentences and relevance to the context. It surpasses other baseline models, confirming the effectiveness of the generative dialogue system developed in this research.

4.4 Relative Position Encoding Performance Verification

To test whether the relative position coding can slow down the weakness of long-distance information, this segment selects the label data set and test data set of multi-round dialogue test data set STC for experiments.

In this section, we choose the dialogue system implemented in segment 3.1 as baseline. Then, multiple models are used for comparative experiments. Finally, the two optimization methods of relative position coding in this paper and word-character fusion [62] embedding are applied to the generative dialogue system at the same time, and compared

Table 6 Model generates labels under different sequence lengths and batch sizes

	Maxlen	Batch size	Loss	ACC
1	128	16	1.86	63.4%
2	256	16	1.78	64.6%
3	256	32	1.75	64.8%
4	512	16	1.66	66.2%

with the baseline model, as shown in Table 5.

First, in the model of this section using relative position coding, the results under different conditions are verified by increasing the sequence length and changing the size of batch size. As shown in Table 6

At the same time, the attentional decline trend images at different relative distances are expressed. The attentional decline results using different θ_t are shown in Figs. 9 and 10 below.

except $\theta_t = t$ is abnormal and intersects with the x -axis, other trends are basically the same. The power function decreases faster in a short distance, while the exponential function decreases faster in a long distance. Therefore, choosing $\theta_t = 10000^{-t}$ is a compromise.

Through the comparative experiments of several models, the objective evaluation index experimental results as shown

Fig. 9 Attentional decline results of different θ_t (short distance trend)

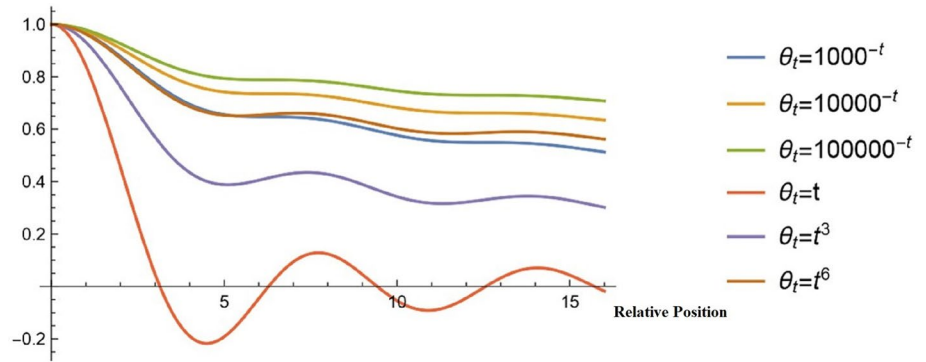


Fig. 10 Attentional decline results of different θ_t (long distance trend)

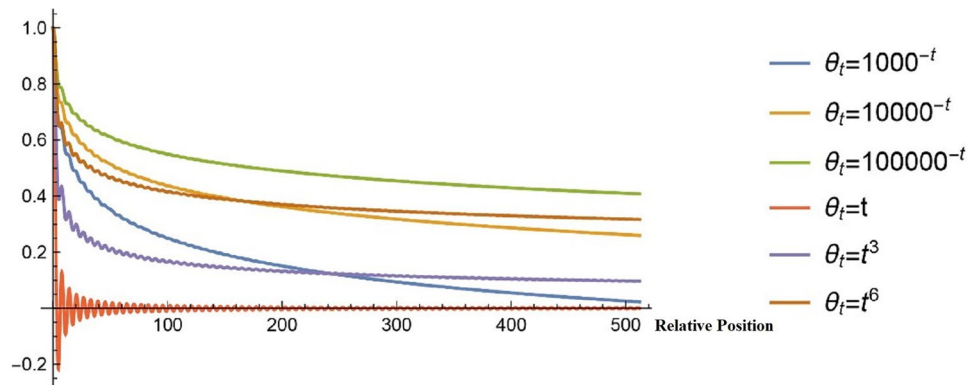


Table 7 Objective evaluation index experimental results

Models	PPL	BLEU-2	BLEU-4	Greedy Matching	Embedding Average
Baseline	19.83	6.63	3.20	65.89	78.94
Char-word	19.79	6.72	3.81	66.34	84.12
Model of this paper	18.56	6.90	4.12	66.28	86.13

Table 8 Labor evaluation index experiment results

Models	PPL	BLEU-2	BLEU-4	Greedy Matching	Embedding Average
Baseline	19.83	6.63	3.20	65.89	78.94
Char-word	19.79	6.72	3.81	66.34	84.12
Model of this paper	18.56	6.90	4.12	66.28	86.13

in Table 7 are obtained. The experimental results of subjective evaluation indicators are shown in Table 8.

It can be seen from the experimental results that the final dialogue system using word-character fusion Embedding and relative position coding achieves the best results in all indicators.

5 Discussion

Through an analysis and comparison of the labels generated by the models under varying sequence lengths and batch sizes, some valuable insights can be gleaned. Specifically, it becomes evident that increasing the length of the sequence

leads to improved accuracy and smaller loss values. It indicates that relative position encoding has better processing power for long text sequences compared to absolute position encoding, which aligns with the anticipated outcomes of the design involving relative position coding.

In terms of objective performance indicators, the model presented in this paper outperforms the baseline and character models in metrics, such as bleu-2, bleu-4, and embedding average. In addition, it outperforms the baseline model in greedy matching and perplexity (ppl), but lags somewhat behind in char word.

Although the model performs admirably in terms of grammatical and semantic coherence during manual evaluation, there is still a definite discrepancy in terms of contextual relevance and the amount of information when compared to the reference response. This emphasizes the significance of future research projects focused on context analysis and maintaining logical coherence in text generation.

These findings also demonstrate relative position coding's improved ability to handle lengthy text sequences as compared to absolute position coding. This observation underscores the effectiveness of the optimization approach employed in this study. In summary, this work showcases the proposal to incorporate relative position information into the self-attention formula within the transformer module, thereby enhancing long-distance attention mechanisms.

6 Conclusions

The main focus of this study is to theoretically present a design framework for a generative dialogue system based on the transformer architecture tailored for Chinese text. The use of transformer technology serves as the foundational framework. To address the limitation of unidirectional generation in language sequences and enable bidirectional access to contextual information within input sentences, the application of partial masking is introduced.

The study also introduces training and optimization techniques for the dialogue system, including teacher forcing and beam search, along with model pretraining on the LCCC data set. Comparative analyses are conducted against various baseline models, such as Attn-Seq2Seq, transformer, and GPT-chatbot to validate the effectiveness of this dialogue system in generating Chinese generative dialogues.

Subsequently, the paper addresses the issue of text length limitations associated with common absolute position encoding. Building upon relative position encoding, the paper proposes a novel technique for relative position encoding tailored for Chinese text. In experiments, the transformer-based Chinese text generation dialogue model developed in this paper is used as the baseline model. The test data set in the short text dialogue corpus STC

released by Huawei's Noah's Ark Laboratory was selected as the test data for the research task. Model performance is evaluated using both absolute and relative position encoding. Experimental results demonstrate the feasibility of enhancing the system's ability to mitigate the phenomenon of long-distance information decay by introducing relative position encoding. Modification of the self-attention calculation formula within the transformer module, by incorporating relative position information to replace the absolute position encoding in the embedding layer, results in enhanced long-range attention capabilities.

In summary, this study's primary contributions lie in offering a theoretical framework for designing a generative dialogue system for Chinese text, and it introduces a novel approach for relative position encoding to address text length limitations. Experimental findings support the effectiveness of this approach in mitigating long-distance information decay, achieved through adjustments to the self-attention mechanism within the transformer module.

Acknowledgements The authors wish to thank the anonymous reviewers for their helpful comments.

Author contributions Conceptualization: WZ and LY; methodology: GG, RW and JT; formal analysis and investigation: XL, SL and ZY; writing—original draft preparation: LY, GG, JT and ZY; writing—review and editing: WZ, LY and JT; software: XL, GG and ZY; data curation: GG and RW; visualization: XL; resources: SL; supervision: WZ; project administration: LY; funding acquisition: WZ. All authors have read and approved the final manuscript.

Funding Supported by Sichuan Science and Technology Program (2021YFQ0003, 2023YFSY0026, 2023YFH0004).

Data availability The data sets presented during the current study are available from the corresponding author on reasonable request.

Declarations

Conflict of interest The authors declare that they have no relevant financial or non-financial interests to disclose.

Employment Not applicable.

Ethical approval The Ethic statement is not applicable. This study does not include any animal or human studies.

Informed consent statement Not applicable.

Institutional review board statement Not applicable.

Consent for publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in

the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Mateju, L., Griol, D., Callejas, Z., Molina, J.M., Sanchis, A.: An empirical assessment of deep learning approaches to task-oriented dialog management. *Neurocomputing* **439**(June), 327–339 (2021). <https://doi.org/10.1016/j.neucom.2020.01.126>
- Ni, J., Young, T., Pandelea, V., Xue, F., Cambria, E.: Recent advances in deep learning based dialogue systems: a systematic survey. *Artif. Intell. Rev.* **56**(4), 3055–3155 (2023). <https://doi.org/10.1007/s10462-022-10248-8>
- Lauriola, I., Lavelli, A., Aiolli, F.: An introduction to deep learning in natural language processing: models, techniques, and tools. *Neurocomputing* **470**(January), 443–456 (2022). <https://doi.org/10.1016/j.neucom.2021.05.103>
- Zhu X (2022) “RNN Language Processing Model-Driven Spoken Dialogue System Modeling Method.” Edited by Xin Ning. *Computational Intelligence and Neuroscience 2022* (February): 1–9. <https://doi.org/10.1155/2022/6993515>.
- Park, Y., Ko, Y., Seo, J.: BERT-based response selection in dialogue systems using utterance attention mechanisms. *Expert Syst. Appl.* **209**(December), 118277 (2022). <https://doi.org/10.1016/j.eswa.2022.118277>
- Junaid T, Sumathi D, Sasikumar AN, Suthir S, Manikandan J, Rashmita K, Kuppusamy PG, Janardhana Raju M (2022) A comparative analysis of transformer based models for figurative language classification. *Comput Electr Eng* **101** (July): 108051. <https://doi.org/10.1016/j.compeleceng.2022.108051>
- Li, J., Joe Qin, S.: Applying and dissecting LSTM neural networks and regularized learning for dynamic inferential modeling. *Comput. Chem. Eng.* **175**(July), 108264 (2023). <https://doi.org/10.1016/j.compchemeng.2023.108264>
- Sherstinsky, A.: Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D* **404**(March), 132306 (2020). <https://doi.org/10.1016/j.physd.2019.132306>
- Weerakody, P.B., Wong, K.W., Wang, G.: Policy gradient empowered LSTM with dynamic skips for irregular time series data. *Appl. Soft Comput.* **142**(July), 110314 (2023). <https://doi.org/10.1016/j.asoc.2023.110314>
- Zhang, X., Shi, J., Yang, M., Huang, X., Usmani, A.S., Chen, G., Jianmin, Fu., Huang, J., Li, J.: Real-time pipeline leak detection and localization using an attention-based LSTM approach. *Process. Saf. Environ. Prot.* **174**(June), 460–472 (2023). <https://doi.org/10.1016/j.psep.2023.04.020>
- Sutskever I, Vinyals O (2014) Sequence to sequence learning with neural networks. *Adv Neural Inform Process Syst*
- Bahdanau D, Cho K, Bengio Y (2014) Neural machine translation by jointly learning to align and translate. *Computer Science*
- Li, J., Chen, R., Huang, X.: A sequence-to-sequence remaining useful life prediction method combining unsupervised LSTM encoding-decoding and temporal convolutional network. *Meas. Sci. Technol.* **33**(8), 085013 (2022). <https://doi.org/10.1088/1361-6501/ac632d>
- Liang, Z., Junping, Du., Li, C.: Abstractive social media text summarization using selective reinforced Seq2Seq attention model. *Neurocomputing* **410**(October), 432–440 (2020). <https://doi.org/10.1016/j.neucom.2020.04.137>
- Britz D, Goldie A, Luong M-T, Quoc L (2017) Massive Exploration of Neural Machine Translation Architectures. *arXiv*
- Chorowski J, Bahdanau D, Serdyuk D, Cho K, Bengio Y (2015) Attention-Based Models for Speech Recognition. *ArXiv.Org*. June 24, 2015
- Shen, Y.: Bionic communication network and binary pigeon-inspired optimization for multiagent cooperative task allocation. *IEEE Trans. Aerosp. Electron. Syst.* **58**(5), 3946–3961 (2022). <https://doi.org/10.1109/TAES.2022.3157660>
- Lv, H., Chen, J., Pan, T., Zhang, T., Feng, Y., Liu, S.: Attention mechanism in intelligent fault diagnosis of machinery: a review of technique and application. *Measurement* **199**(August), 111594 (2022). <https://doi.org/10.1016/j.measurement.2022.111594>
- Shi, Q., Fan, J., Wang, Z., Zhang, Z.: Multimodal channel-wise attention transformer inspired by multisensory integration mechanisms of the brain. *Pattern Recogn.* **130**(October), 108837 (2022). <https://doi.org/10.1016/j.patcog.2022.108837>
- Zhang, X., Yawen, Wu., Zhou, P., Tang, X., Jingtong, Hu.: Algorithm-hardware co-design of attention mechanism on FPGA devices. *Acm Trans Embedded Comput Syst* **20**(5), 71 (2021). <https://doi.org/10.1145/3477002>
- Ni, J., Huang, Z., Chang, Yu., Lv, D., Wang, C.: Comparative convolutional dynamic multi-attention recommendation model. *Ieee Trans Neural Netw Learn Syst* **33**(8), 3510–3521 (2022). <https://doi.org/10.1109/TNNLS.2021.3053245>
- Chen, J., He, Ye.: A novel u-shaped encoder–decoder network with attention mechanism for detection and evaluation of road cracks at pixel level. *Comput-Aid Civ Infrastruct Eng* **37**(13), 1721–1736 (2022). <https://doi.org/10.1111/mice.12826>
- Du, S., Li, T., Yang, Y., Horng, S.-J.: Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomputing* **388**(May), 269–279 (2020). <https://doi.org/10.1016/j.neucom.2019.12.118>
- Feng, L., Zhao, C., Sun, Y.: Dual attention-based encoder–decoder: a customized sequence-to-sequence learning for soft sensor development. *IEEE Trans Neural Netw Learn Syst* **32**(8), 3306–3317 (2021). <https://doi.org/10.1109/TNNLS.2020.3015929>
- Mikolov T (2012) Statistical language models based on neural networks. PhD thesis, Brno University of Technology
- Schuster, M., Paliwal, K.: Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45**(11), 2673–2681 (1997)
- Sundermeyer, M., Schluter, R.: From feedforward to recurrent LSTM neural networks for language modeling. *IEEE/ACM Trans Audio Speech Lang Process* **23**(3), 517–529 (2015)
- Zhu, S., Cheng, X., Sen, Su.: Knowledge-based question answering by tree-to-sequence learning. *Neurocomputing* **372**(January), 64–72 (2020). <https://doi.org/10.1016/j.neucom.2019.09.003>
- Liu T, Wang K, Sha L, Chang B, Sui Z Table-to-text generation by structure-aware Seq2seq learning. proceedings of the AAAI conference on artificial intelligence 32 <https://doi.org/10.1609/aaai.v32i1.11925> (2018).
- Vaswani A, Shazeer N, Parmar N et al. (2017) Attention is all you need. In: *Advances in neural information processing systems*, pages 5998–6008
- Niu, Z., Zhong, G., Hui, Yu.: A review on the attention mechanism of deep learning. *Neurocomputing* **452**(September), 48–62 (2021). <https://doi.org/10.1016/j.neucom.2021.03.091>
- Qun, He., Wenjing, L., Zhangli, C.: B&Anet: combining bidirectional LSTM and self-attention for end-to-end learning of task-oriented dialogue system. *Speech Commun.* **125**(December), 15–23 (2020). <https://doi.org/10.1016/j.specom.2020.09.005>
- Beltagy Iz, Matthew EP, Cohan A (2020) Longformer: The Long-Document Transformer. *arXiv*.
- Shan, W., Huang, D., Wang, J., Zou, F., Li, S.: Self-attention based fine-grained cross-media hybrid network. *Pattern Recogn.*

- 130(October), 108748 (2022). <https://doi.org/10.1016/j.patcog.2022.108748>
35. Dufter, P., Schmitt, M., Schütze, H.: Position information in transformers: an overview. *Comput. Linguist.* **48**(3), 733–763 (2022). https://doi.org/10.1162/coli_a_00445
 36. Yida W, Ke P, Zheng Y, Huang K, Jiang Y, Zhu X, Huang M (2020) A large-scale chinese short-text conversation dataset. In: Paper presented at the Natural Language Processing and Chinese Computing, Cham. https://doi.org/10.1007/978-3-030-60450-9_8
 37. Abdalla, H.I., Amer, A.A., Amer, Y.A., et al.: Boosting the item-based collaborative filtering model with novel similarity measures. *Int J Comput Intell Syst* **16**, 123 (2023). <https://doi.org/10.1007/s44196-023-00299-2>
 38. Amer AA, Abdalla HI, Nguyen L (2021) Enhancing recommendation systems performance using highly-effective similarity measures. *Knowl-Based Syst* **217**: 106842. <https://doi.org/10.1016/j.knosys.2021.106842>
 39. Liu, Z., Liu, H., Jia, W., Zhang, D., Tan, J.: A multi-head neural network with unsymmetrical constraints for remaining useful life prediction. *Adv. Eng. Inform.* **50**(October), 101396 (2021). <https://doi.org/10.1016/j.aei.2021.101396>
 40. Reza, S., Ferreira, M.C., Machado, J.J.M., João, M.R., Tavares, S.: A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks. *Expert Syst. Appl.* **202**(September), 117275 (2022). <https://doi.org/10.1016/j.eswa.2022.117275>
 41. Zhang L, Wang C-C, Chen X (2022) Predicting Drug-target binding affinity through molecule representation block based on multi-head attention and skip connection. *Briefings Bioinform* **23**(6): bbac468. <https://doi.org/10.1093/bib/bbac468>.
 42. Zheng, W., Yin, L.: Characterization inference based on joint-optimization of multi-layer semantics and deep fusion matching network. *PeerJ Comput Sci* **8**(April), e908 (2022). <https://doi.org/10.7717/peerj-cs.908>
 43. Zheng, W., Zhou, Yu., Liu, S., Tian, J., Yang, Bo., Yin, L.: A deep fusion matching network semantic reasoning model. *Appl. Sci.* **12**(7), 3416 (2022). <https://doi.org/10.3390/app12073416>
 44. Atta, E.A., Ali, A.F., Elshamy, A.A.: A modified weighted chimp optimization algorithm for training feed-forward neural network Edited by Kathiravan Srinivasan. *PLoS ONE* **18**(3), e0282514 (2023). <https://doi.org/10.1371/journal.pone.0282514>
 45. Ma, Z., Zheng, W., Chen, X., Yin, L.: Joint embedding VQA model based on dynamic word vector. *PeerJ Computer Science* **7**(March), e353 (2021). <https://doi.org/10.7717/peerj-cs.353>
 46. Zong, Yi., Pan, E.: A SOM-based customer stratification model. *Wirel. Commun. Mob. Comput.* **2022**(March), e7479110 (2022). <https://doi.org/10.1155/2022/7479110>
 47. Gehring J, Auli M, Grangier D, Yarats D, Dauphin YN. (2017) Convolutional Sequence to Sequence Learning. In: Proceedings of the 34th international conference on machine learning, 1243–52. PMLR.
 48. Liu X, Yu H-F, Dhillon I, Hsieh C-J (2020) Learning to encode position for transformer with continuous dynamical model. In: Proceedings of the 37th international conference on machine learning, 6327–35. PMLR.
 49. Abdalla HI, Amer AA (2022) On the integration of similarity measures with machine learning models to enhance text classification performance. *Inform Sci* **614**: 263–288. <https://doi.org/10.1016/j.ins.2022.10.004>
 50. Abdalla HI, Amer AA, Ravana SD (2023) BoW-based neural networks vs. cutting-edge models for single-label text classification. *Neural Comput Appl* **35**(27): 20103–20116. <https://doi.org/10.1007/s00521-023-08754-z>
 51. Shang L, Lu Z, Li H (2015) Neural responding machine for short-text conversation. *arXiv*. <https://doi.org/10.48550/arXiv.1503.02364>.
 52. Vinyals O, Le Q (2015) A neural conversational model. *arXiv*.
 53. Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the association for computational linguistics, 311–18. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics. <https://doi.org/10.3115/1073083.1073135>.
 54. Corley C, Mihalcea R (2005) Measuring the Semantic Similarity of Texts. In: Proceedings of the ACL workshop on empirical modeling of semantic equivalence and entailment, 13–18. Ann Arbor, Michigan: Association for Computational Linguistics.
 55. Lintean M, Rus V (2012) Measuring semantic similarity in short texts through greedy pairing and word semantics. In: Proceedings of the twenty-fifth international FLAIRS conference, Marco Island, FL, USA, 23–25 May
 56. Yadav, S., Kaushik, A.: Do you ever get off track in a conversation? the conversational system’s anatomy and evaluation metrics. *Knowledge* **2**(1), 55–87 (2022). <https://doi.org/10.3390/knowledge2010004>
 57. Wieting J, Bansal M, Gimpel K, Livescu K (2016) Towards Universal Paraphrastic Sentence Embeddings. *arXiv*.
 58. Zhong, S.-H., Liu, P., Ming, Z., Liu, Y.: How to evaluate single-round dialogues like humans: an information-oriented metric. *IEEE/ACM Trans Audio Speech Lang Process* **28**, 2211–2223 (2020). <https://doi.org/10.1109/TASLP.2020.3003864>
 59. Zhang, C., Lee, G., D’Haro, L.F., Li, H.: D-score: holistic dialogue evaluation without reference. *IEEE/ACM Trans Audio Speech Lang Process* **29**, 2502–2516 (2021). <https://doi.org/10.1109/TASLP.2021.3074012>
 60. Oluwatobi O, Mueller E (2020). DLGNet: A transformer-based model for dialogue response generation. In: Proceedings of the 2nd workshop on natural language processing for conversational AI
 61. Zhang Y, Sun S, Galley M, Chen Y-C, Brockett C, Gao X, Gao J, Liu J, Dolan B (2019) Dialogpt: Large-scale generative pre-training for conversational response generation. *arXiv preprint arXiv:1911.00536*.
 62. Luo J, Zou X, Hou M (2022) A novel character-word fusion chinese named entity recognition model based on attention mechanism. In: 2022 IEEE 5th international conference on computer and communication engineering technology (CCET)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.