Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Research on Intelligent Interaction Design for Enhancing Multi-Scenario English Learning Using ChatGPT and Deep Reinforcement Learning

Ruijiao You[1,✉]

[1] *School of General Education, Zhangzhou Health Vocational College, Zhangzhou, Fujian, 363000, China*

ABSTRACT

Human-computer interaction scenarios have a broad prospect in the field of English learning. In this paper, a human-computer dialogue interaction system for English learning scenarios is designed based on deep reinforcement learning and artificial intelligence interaction technology. Firstly, a speech enhancement method based on collaborative recurrent network is proposed to optimize the speech analysis module. On this basis, we design the framework of human-computer interaction system, and construct a human-computer dialogue interaction system for English learning scenarios that contains three modules: natural language understanding (NLU), knowledge retrieval enhancement, and natural language generation (NLG), in which knowledge retrieval enhancement utilizes ChatGPT for document reordering design. In the speech enhancement simulation experiments, the mean value of network congestion for the speech enhancement method designed in this paper is 0.073, which achieves at least 50% performance improvement, reduces speech distortion and optimizes the signal-to-noise ratio at the same time. The system is experimentally analyzed for two tasks, conversation state tracking and conversation reply generation, and outperforms the baseline model on both tasks. Finally, a subjective evaluation is conducted, and the system in this paper scores 3.766, which is obviously a smoother human-computer interaction experience, and the English learning interaction experience has a greater advantage compared with the other methods. This paper provides innovative ideas and feasible methods for combining cutting-edge information technology with interactive English teaching.

Keywords: speech enhancement, recurrent neural network, natural language understanding, English learning

1. Introduction

Multi-scenario English learning is a method of learning English by simulating real-life scenarios. The advantage of this method is that it can help learners master English in real contexts and improve the

✉ Corresponding author.
E-mail address: you20220321@163.com (R. You).
Received 03 March 2024; Revised 10 May 2024; Accepted 30 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-221

practical use of language [1-4]. Multi-scene English learning has the following characteristics: authenticity: simulating real-life scenarios, allowing learners to learn English in real contexts, and improving the practical use of language [5-6]. Practicality: learners can learn English through scenes, learn common English expressions in real life, and improve their English communication skills in life and work [7-8]. Interestingness: Through interesting scene design, learners' interest in learning can be increased, making learning more interesting. Integrating intelligent interaction into scene-based English learning is important for improving English proficiency [9-11].

Intelligent interaction design for multi-scene learning is a learning environment with human-computer interaction created with the help of emerging technologies such as Artificial Intelligence, through the design and development of interactive systems, so that they can understand and interpret the needs of teachers and students, and communicate and interact with the users in an appropriate way [12-15]. Intelligent interaction design is a discipline involving human-computer interaction, user experience and artificial intelligence, aiming at realizing efficient, flexible and intelligent communication and interaction between humans and machines [16-18]. For multi-scenario English learning, intelligent interaction design is not only a reform and innovation of the traditional teaching method, but also a necessary way for the development of the times. Driven by this design, English learning will achieve excellent results [19-22].

In this paper, we construct a human-computer dialogue interaction system for English learning scenarios based on deep reinforcement learning. The AR parameter estimation model of the speech signal is first constructed based on the generalized least absolute deviation method, and the deep recurrent Q-network is used to solve the model. Then according to the required parameters, the speech signal data are sequentially reduced by Kalman filter recurrent network. On this basis, the construction of a human-computer interaction system for multi-scene English learning is realized by combining ChatGPT. The natural language understanding module realizes extracting semantic slots from the user-input dialogue and analyzing the current dialogue scenario. The knowledge retrieval module realizes to retrieve a subset of knowledge related to the current conversation from an external knowledge base based on the semantic slot information and the conversation scenario, and enhances it with ChatGPT. The natural language generation module generates a natural language sentence to complete the interaction with the user. After the system is built, simulation experiments and subjective evaluation are conducted.

2. Intelligent interaction technology based on deep reinforcement learning

2.1. Phonological Enhancement for English Learning

Speech enhancement is a digital signal processing technique that extracts useful speech signals by suppressing and reducing the effect of environmental noise to improve the speech received by the robot [23].

Consider the noise and speech signals as autoregressive AR models driven by Gaussian white noise with appropriate orders, respectively. Let the order of the AR model $s(i)$ of the original speech signal be p, the number of sample points be N, and the original speech sequence be $\{s(i)\}_1^N$, $s(i)$ which can be expressed as:

$$s(i)=\sum_{j=1}^{p}a_j s(i-j)+u(i) \tag{1}$$

where a_j is the linear prediction (LPC) coefficient, which is also an AR parameter.

Let the observed noise of the simulated input be $v(i)$ experimentally usually a Gaussian white noise with zero mean and variance σ_e^2. The AR model of the noise-contaminated speech signal $y(i)$ can be expressed as follows:

From the noisy signal $y(i)$, the original speech signal $s(i)$ is extracted optimally:

$$y(i) = s(i) + v(i) \tag{2}$$

The process of estimating the value $s(i)$, i.e., the process of speech enhancement.

2.2. Recursive Q-networks based on generalized least absolute deviation parameter estimation

2.2.1. Generalized minimum absolute deviation parameter estimation. The known AR model with noise signal $y(t)$ can be expressed as follows:

$$y(t) = y_t^{\mathrm{T}} a - n(t) \tag{3}$$

where t is the position of the observed speech signal in the sample point ensemble, $t \in [1, \cdots, N]$. $y_i = [y(t-1), \cdots, y(t-p)]^{\mathrm{T}}$ denotes the current observed speech vector, $a = [a_1, \cdots, a_p]$ is the unknown AR parameter, and $n(t) = u_t^{\mathrm{T}} a - u(t) - v(t)$ is the noise vector, where $u_t = [u(t-1), \cdots, u(t-p)]^{\mathrm{T}}$. Transforms Eq. (3) into matrix-vector form, and $y = [y(1), \cdots, y(N)]^{\mathrm{T}}, n = [n(1), \cdots, n(N)]^{\mathrm{T}}$, then the transformed Eq:

$$\dot{Y}a = y + n \tag{4}$$

In order to obtain more accurate parameter estimates, a noise error vector is introduced, and the noise error vector is optimized at the same time as the parameter vector. Let the upper and lower bounds of the noise vector $n(t)$ be $n^+(t)$ and $n^-(t)$, $n(t) \le n(t) \le n^+(t)$, respectively, then the noise vector error can be defined as:

$$\Omega_y = \{z \in R^N \mid \gamma_1 l \le z \le \gamma_1 h\} \tag{5}$$

where $l = \left[n^-(t), \cdots, n^-(N)\right], h = \left[n^+(t), \cdots, n^+(N)\right]$, and $l \le n \le h$. γ_1, γ_2 is the proportionality constant, z is the noise error vector, and $n \in \Omega_y$ when the proportionality constant is set to 1.

The generalized absolute minimum deviation optimization problem model is constructed by combining the linear equation (4) and the noise error set Ω_γ.

2.2.2. Deep Recursive Q-Networks. A recurrent network is used as the algorithm for solving the generalized absolute minimum deviation optimization problem in this paper.

The recurrent neural network is first used for solving the problem:

$$\frac{da}{dt} = -\lambda \left\{ Y^T e - Y^T (e - g^1(e + Ya - y - z)) \right\} \tag{6}$$

$$\begin{aligned} \frac{de}{dt} &= -\lambda \left\{ e + YY^T e - g^{-1}(e + Ya - y - z) - (z - g^2(z + e)) \right\} \\ \frac{dz}{dt} &= -\lambda \left\{ z - g^2(z + e) + e - g^1(e + Ya - y - z) \right\} \end{aligned} \tag{7}$$

where $a \in R^p$, $e \in R^N$ and $z \in R^N$ are state vectors, respectively. λ is the design constant. g^1 and g^2 are defined as follows:

$$g^1(z_i) = \begin{cases} -1 & z_i < -1 \\ z_i & -1 \le z_i \le 1 \\ 1 & z_i > 1 \end{cases} \tag{8}$$

$$g^2(z_i) = \begin{cases} l_i & z_i < l_i \\ z_i & l_i \le z_i \le h_i \\ h_i & z_i > h_i \end{cases} \tag{9}$$

Eq. (7) represents each of the three state models that constitute the recurrent neural network.

It is known that the optimal Q-value of the model in state s is defined as $Q^*(s,a)=\max_v Q^\pi(s,a)$. By means of the optimal Q-value, the optimal action of the model in state s can be calculated by $\arg\max_a Q(s,a)$. The optimal policy is calculated based on the optimal action of the model in each state, and the method of updating the Q value of the model from state-action (s,a) to the next state-action (s',a') can be calculated as follows:

$$Q(s,a):=Q(s,a)+\alpha(r+\gamma\max_{a'}Q(s',a')-Q(s,a)) \tag{10}$$

The complexity of the model is reduced by finding the optimal weighting factors and setting the network parameters to solve the Q-value $Q(s,a\,|\,w)$ for a given state input instead of the original Q learning.

Bellman residuals are used as the error calculation method for QDN, and the loss function of the model can be expressed as:

$$L_i(w_i)=E_{(s,a,r,s')\sim D}\left[(r+\gamma\max_{a'}Q(s',a';w_i)-Q(s,a;w_i))^2\right] \tag{11}$$

All historical states, actions, and expectations are stored to the experience pool and sampled in the experience pool according to certain rules at each update. The final operation flow of the deep recurrent network improved based on the Q-learning concept is shown in Fig. 1.

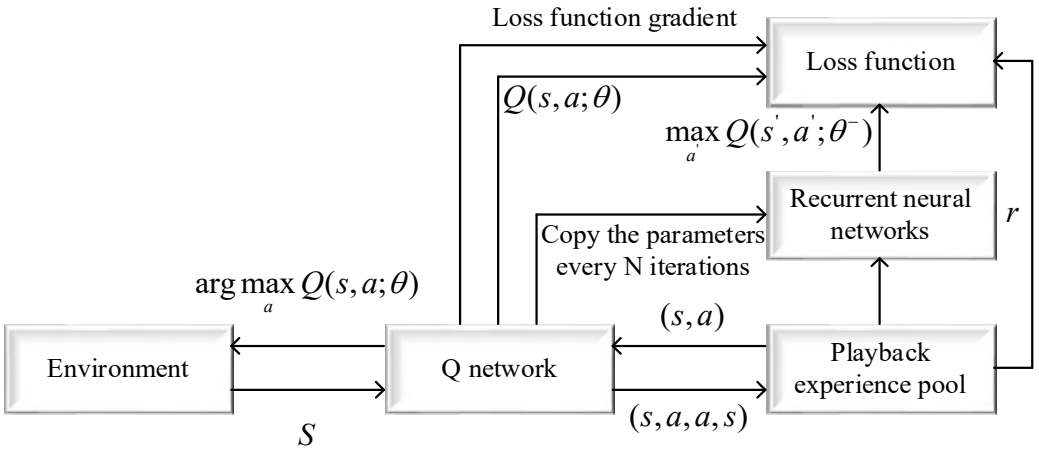


Fig. 1. Specific frame diagram of deep recursive Q network

2.3. Recurrent neural network based on Kalman filtering

Kalman filtering divides the original language model into a speech model and a noisy model, and the enhancement of the robot's effective speech is realized by the optimal estimation of the original speech in the sense of the minimum mean square error.

It is known that the original speech model and the model with noisy speech are represented separately in the time domain as follows:

$$\begin{cases} s(n)=\sum_{j=1}^{p}a_j s(n-j)+u(n) \\ y(n)=s(n)-v(n) \end{cases} \tag{12}$$

where $s(n)$ is the n nd signal in the original speech sample. $y(n)$ is the n th noisy speech signal in the observed speech sample. Converting Eq. (12) to a canonical state-space matrix equation so that it can be used directly in subsequent calculations using Kalman filtering, the converted equation can be expressed as:

$$\begin{cases} x(n) = F(n)\,x(n-1) + Gu(n) \\ y(n) = Hx(n) + v(n) \end{cases} \tag{13}$$

where $x(n) = [s(n-p+1)s(n-p+2)\cdots s(n)]^{\mathrm{T}}$. F is the order transformation matrix.

The error covariance matrix is constructed based on the theory of standard Kalman filtering algorithm with Kalman filtering gain $K(n)$:

$$P(n) = \left[\, I - K(n)H \right] P(n \mid n-1) \tag{14}$$

where I is the unit matrix. $P(n \mid n-1)$ is the a priori error covariance matrix, $P(n \mid n-1) = FP(N-1)F^T + GG\delta_n^2$, δ_n^2 are the AR parameter estimates.

Therefore, when the initial state conditions are given and the AR parameter estimates and the associated noise statistical properties are known, the valuation of the speech signal at moment n can be calculated as follows:

$$s(n) = \dot{H}x(n) \tag{15}$$

Given the initial conditions, the Kalman filter is a discrete neural network with a recursive structure, which is presented as shown in Fig. 2.

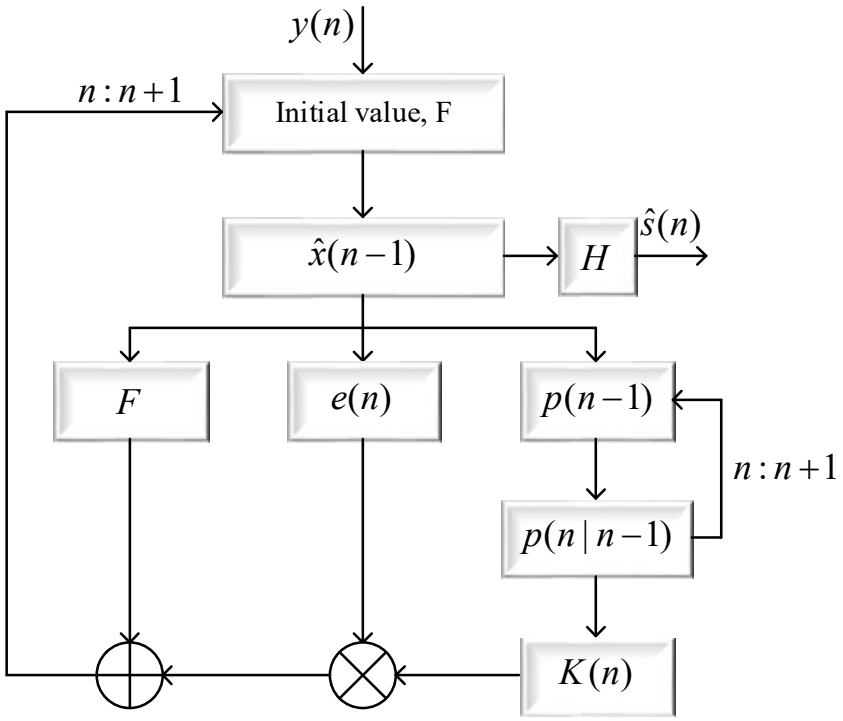


Fig. 2. Kalman filter recursive network structure diagram

2.4. English Speech Enhancement Based on Collaborative Recurrent Neural Networks

Collaborative recurrent neural networks can also be divided into two sub-modules, recurrent Q-networks based on generalized least absolute deviation parameter estimation and recurrent neural networks based on Kalman filtering, which are used to solve the two sub-problems of speech enhancement.

Given a noisy speech sequence $y(i),(i = 1,\cdots,m)$, the speech data is processed in frames with frame length N, and the total number of frames after processing is L. The speech reduction process for frame i is as follows:

1) The first neural network in the collaborative neural network is used to solve the AR parameters of

the speech signal model in frame i [24].

2) According to the solved AR parameters, the second neural network is used to sequentially restore the N speech signal data in frame i.

3) Compare the sizes of i and L, if $i < L$, return to (1), and restore the speech data in frame $i+1$ according to the above method until $i = L$, stop the restoration, and the researcher will count all the restoration results.

The specific structure of the proposed collaborative recurrent neural network based speech enhancement method is shown in Fig. 3:

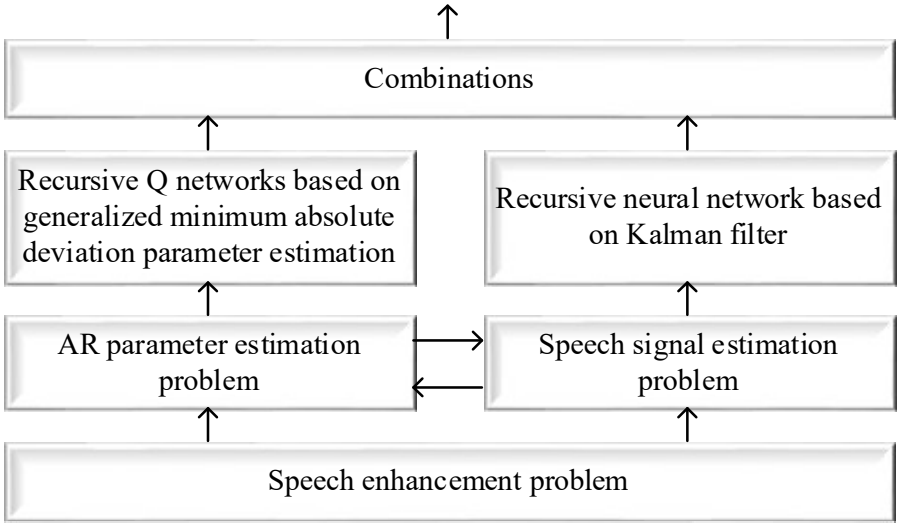


Fig. 3. Architecture of cooperative recurrent neural network for speech enhancement

2.5. Human-computer interaction system construction

2.5.1. Hardware. The hardware part of the system's voice interaction system mainly consists of three main parts: voice acquisition, portable computer and voice output device. Among them, the voice acquisition equipment mainly includes voice sensors, sound, sound card, etc. The voice information collected by the sensors is converted into digital signals by the sound card after the sound method and input into the computer for processing. The portable computer, through software, analyzes and converts the voice signal into command information that can be recognized by the system, and the control platform searches for suitable answers in the Q&A library and converts them into voice output commands, which are output through the speakers. The hardware part of the dialog system voice interaction system, the specific design is shown in Figure 4.

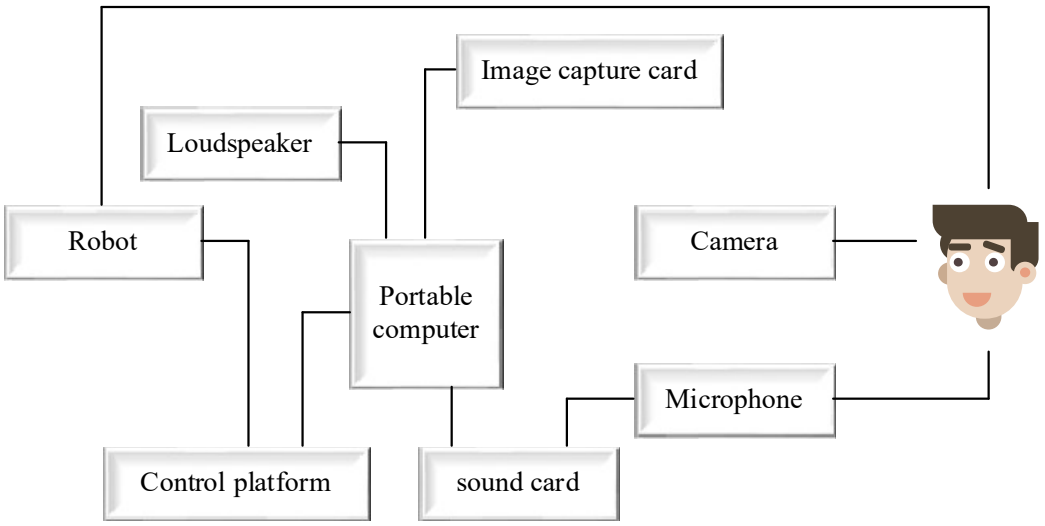


Fig. 4. Structure of the speech interaction system of the dialogue robot

2.5.2. Software component. The software part of the dialogue system voice interaction system is realized based on the computer platform, which is mainly divided into three modules: voice analysis, voice recognition and voice synthesis. The voice analysis module in the dialogue system human-computer interaction system is optimized using the proposed voice enhancement method, and the specific framework of the optimized system voice interaction system is shown in Figure 5.

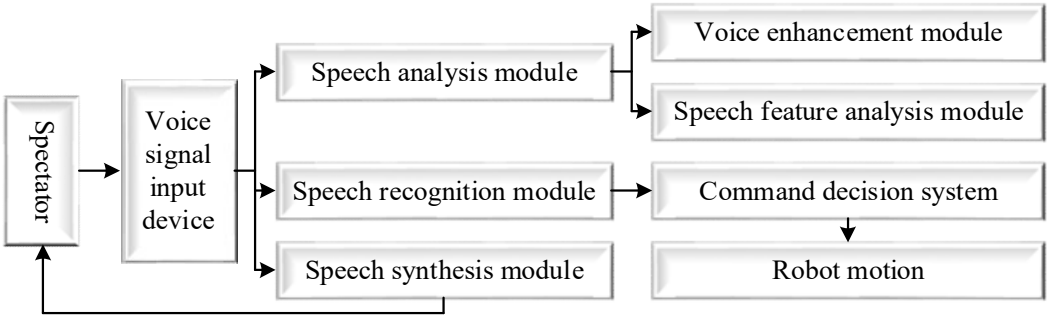


Fig. 5. Specific framework of speech interaction system of dialogue robot

The speech data that has been converted into a digital signal by the sound card is input into the speech analysis module in the computer, and the front-end processing of the speech signal is carried out through the steps of preprocessing, frame-splitting and windowing, end-point detection, and estimation of the fundamental tone period, and the characteristic parameters of the speech signal are extracted. The extracted speech feature signals are recognized and processed by the speech recognition software, and the recognized speech commands are classified and processed according to their functions.

3. Multi-scene human-computer dialog interactive system for English learning

3.1. Natural Language Understanding (NLU) Module

The NLU module is the first step in the workflow of the system's framework, which is mainly tasked with named entity recognition and scenario categorization, and is trained using a supervised learning approach.

The NLU module first attaches a special tag character "@Scenery" to the dialog history X at the time of input to refer to the task scenario of the current dialog. Next, the BiGRU network is utilized to

encode the dialog history $X=[x_1,\ldots,x_i,\ldots,x_n,x_@]$ after filling the special marker to obtain the hidden vector representation $H=[h_1,\ldots,h_1\ldots,h_n,h_@]$ corresponding to each input character. Finally, based on the vector representation H the $[h_1,\ldots,h_1\ldots,h_n]$ will be input into the CRF layer for sequence labeling, extract all the attribute values in the dialog history, and input the vector representation corresponding to the special characters $h_@$ into the multilayer perceptual machine for scenario classification. The computational formula is shown below [25].

$$h_i^{forward}=GRU^{forward}(e(x_i),h_{i-1}^{forward}) \tag{16}$$

$$h_i^{backward}=GRU^{backward}(e(x_i),h_{i+1}^{backward}) \tag{17}$$

$$h_i=[h_i^{forward};h_i^{backward}] \tag{18}$$

The above equation shows the process of coding using BiGRU network, where $e()$ denotes the Embedding layer, $e(x_i)$ that is the word vector of x_i, which is specifically initialized using the Glove word vector. $GRU^{forward}$ denotes the left-to-right coding of the sequence, and $h_i^{forward}$ is the coding result of all the words before the synthesis of x_i. $GRU^{backward}$ means encoding the sequence from right to left, $h_i^{backward}$ is the encoding result of all words after x_i, and then $h_i^{forward}$ and $h_i^{backward}$ are spliced together as the encoding vector of the current moment i, which takes into account the contextual information of x_i. After obtaining the coding result $H=[h_1,\ldots,h_i\ldots,h_n,h_@]$ of the dialog history X, the calculation is done as follows:

$$Y_{logit}=CRF([h_1,\ldots.,h_n]) \tag{19}$$

$$Scene_{logit}=soft\max(MLP(h_@))] \tag{20}$$

where $Y_{logit}=[\hat{c}_1,\ldots,\hat{c}_n]$ denotes the result of coding vector $[h_1,\ldots,h_n]$ after the CRF layer computation. $Scene_{logit}$ denotes the probability distribution of the dialog scenario. Finally the function of NLU module loss is computed $Loss$.

$$Loss=CrossEntropy(Y_{togit},Y_{target})+CrossEntropy(Scene_{logit},Scene_{target}) \tag{21}$$

Y_{target} and $Scene_{target}$ denote the standard distribution of the cross-entropy loss function for the slot recognition and scenario classification tasks, respectively. When training the NLU module network independently, the input data and labels are existing, so the standard distribution corresponding to the input data can be determined, the cross entropy is a calculation method to measure the degree of difference between the predicted probability distribution and the real standard distribution, the larger the entropy value represents the larger difference between the predicted distribution and the real distribution, and the smaller the entropy value the smaller the difference is, therefore, the selection of the cross entropy as the loss function can make the model to be accurately trained. The cross entropy formula is shown below:

$$CrossEntropy(p,q)=-\sum_{i=1}^{n}p(x_i)\log(q(x_i)) \tag{22}$$

where $p(x)$ denotes the true probability distribution and $q(x)$ denotes the predicted probability distribution.

3.2. Retrieval enhancement model based on ChatGPT

3.2.1. Fine-tuning-based document reordering. Paragraph reordering is a central task in the field of information retrieval (IR). Its core goal is to reorder the list of paragraphs obtained from preliminary retrieval models to improve retrieval accuracy. Fine-tuning and applying pre-trained Large Language Models (LLMs) to the document reordering task has become a mainstream approach. According to the

strategy of fine-tuning, the existing methods can be divided into two main categories: (1) Fine-tuning LLMs as generative models. (2) Fine-tuning LLMs as ranking models.

Fine-tuning LLMs as Sorting Models While fine-tuning LLMs as generative models performs well on some baseline tasks, this is not always the best choice. This is mainly because, first, ideally, the reordering model should generate a numerical relevance score for each query-document pair, rather than simple text labels. Second, the use of ranking losses, such as RankNe, is more suitable for optimizing reordering models than generative losses. While pre-trained models like BERT have been applied to document reordering, the use of seq2seq-based LLMs, such as T5-3B, has not been extensively studied. More recently RankT5 calculates relevance scores directly for query-document pairs and uses "pairwise" or "listwise" loss to optimize rankings.

3.2.2. Hint-based document reordering. Despite good results, supervised fine-tuning-based approaches face problems such as high annotation costs, weak generalization to new domains or languages, and limitations in their commercial applications. Recently, with the rapid progress of Large Language Models (LLMs) in the field of Natural Language Processing (NLP), they have been shown to have a superior ability to learn with few/zero samples as well as to execute instructions. As a result, a large amount of research has begun to explore how to utilize cue learning to apply LLMs to sequencing systems. In this section, we focus on two existing large model-based reordering methods that employ query generation and relevance generation strategies, respectively.

The relevance between a query and a paragraph is evaluated by the logarithmic probability that the model generates a query based on the paragraph. For a given query q and paragraph p_i, the correlation score s_i between them can be defined as:

$$s_i = \frac{1}{|q|}\sum_t \log p(q_t \mid q_{<t}, p_i, I_{query}) \tag{23}$$

where $|q|$ denotes the number of tokens for query q, q_t is the tth token for query q, and I_{query} is the instruction. After that, the paragraphs are sorted according to the relevance score s_i.

When LLMs are instructed to output "yes", it indicates that the query is relevant to the paragraphs. On the contrary, when "No" is output, the query is not relevant to the paragraph. The relevance score s_i is determined based on the probability that the LLMs generate the word "yes" or "no":

$$s_i = \begin{cases} 1 + p(\text{Yes}), & \text{If the output is yes} \\ 1 - p(\text{No}), & \text{If the output is no} \end{cases} \tag{24}$$

where p (yes/no) represents the probability that the LLMs generate a yes or no response. Note that the relevance scores have been normalized to the range [0, 2].

Both methods rely on log-probability outputs from LLM, however, such outputs may not be accessible for some LLMAPIs. For example, OpenAI's Chat Completion API does not provide log probabilities for generating results. In light of this, we next present an innovative directive alignment generation method that simply uses text-based output to reorder paragraphs. We also propose a sliding window strategy to address the maximum length limit of LLMs.

3.2.3. Sort Generation. In this section, we introduce a new approach to alignment generation. In this method, we input a set of paragraphs with unique identifiers into the language model. Examples include [1], [2], etc. Then, we instruct these models to determine the sort order of the paragraphs based on their relevance to the query. For example, the ranking of the output might be presented as [2] > [3] > [1], etc. It is worth noting that the present method determines the ranking of paragraphs directly without additionally calculating the relevance score of each paragraph.

Due to the maximum input length restriction of the language model, we can only arrange a limited number of paragraphs. To solve this problem, we design the sliding window strategy. Suppose the

retrieval model in the first stage returns M paragraph. We use a sliding window strategy to rearrange these passages in back-to-front order. The strategy involves two hyperparameters: window size w and step size s. First, the model aligns the paragraphs between $(M-w)$ and M. Then, each time the window is moved in steps of s, the paragraphs in the range $(M-w-s)$ to $(M-s)$ are rearranged. This process continues until all paragraphs have been rearranged.

3.2.4. Sorting Distillation. Despite the superior capabilities of large language models like GPT-4, their deployment in commercial search systems is too costly and may increase the response time of the search system significantly. In addition, we have found that large language models (LLMs) sometimes produce unstable generation results. To address these issues, we propose to distill the ranking capabilities of a large model into a smaller, more specialized model through model distillation techniques.

In this paper, we propose an innovative distillation method, sorting distillation, which aims to distill ChatGPT's paragraph reordering capability into a smaller, specialized model. Compared to existing distillation methods, our approach directly uses the alignments generated by the larger model as the distillation target, avoiding the need to induce additional inductive biases such as consistency checking or log-probability manipulation.

Consider a query q and a M document fragment obtained through the BM25 retrieval algorithm, denoted $(p_1,\ldots,p_M)$. In the experimental setup of this study, we set $M=20$. We use ChatGPT as a teacher model to predict the ranking $R=(r_1,\ldots,r_M)$ of documents, where $r_i \in [1,2,...,M]$ denotes the ranking of document p_i. For example, if $r_i=3$, it means that p_i is ranked third out of M documents. Further, let us have a specialized ranking model $s_i=f_\theta(q,p_i)$ with parameter θ. This model uses a Cross-Encoder to compute the correlation score s_i between the query q and the document fragment p_i. To optimize this model, we utilize the ranking generated by ChatGPT R and optimize it with RankNet loss:

$$\mathrm{L}_{RankNet}=\sum_{i=1}^{M}\sum_{j=1}^{M}\mathrm{K}_{r_i<r_j}\log(1+\exp(s_i-s_j)) \tag{25}$$

It is worth noting that RankNet is a pairwise loss function that is primarily used to evaluate the accuracy of document fragment sorting. When using the sort R generated by ChatGPT, we can construct a total of $M(M-1)/2$ document fragment pairs for comparison.

Regarding the architecture of the student model, we consider two model structures: the BERT-type model and the GPT-type model.

BERT-type model. We use a cross-coder model based on DeBERTa-large. It uses [SEP] tags to connect queries and text segments, and uses [CLS] tagged representations to estimate correlation scores.

GPT-type model. We use LLaMA-7B with a relevance-generating hint template. It classifies queries and paragraphs as relevant or irrelevant by generating a relevance marker. The relevance score is then defined as the probability of generating a relevance marker.

3.3. Natural Language Generation (NLG) Module

The natural language generation module is the final step in the workflow of the multi-task situational human-computer dialogue system, which is mainly responsible for receiving the inputs of both the candidate knowledge subset and the dialog history x retrieved by the KR module, and generating the system responses.

3.3.1. Dialog encoder. The dialogue encoder takes dialogue history X as input and encodes each word in dialogue X as a vector representation according to a hierarchical encoding approach, while a comprehensive vector representation of dialogue history X is computed. The dialog encoder employs a two-layer bi-directional GRU network for encoding intra-sentence and inter-sentence

information respectively, which are referred to here as sentence-level BiGRU encoder and dialog-level BiGRU encoder. The specific workflow is as follows:

1) First, each word in the dialog history $X=[x_1,...,x_i,...x_n]$ is expanded to $X=[c_1,...,c_i,...c_n]$ based on the dialog role (user or system) and dialog turn, where $c_i=<x_i,u/s,t>$, u denotes a user utterance, s denotes a system reply, "/" denotes an or relation, and t denotes a dialog turn. The expanded conversation history enables the model to capture more information related to the conversation when encoding word sequences.

2) Divide the conversation history x into conversation rounds, where a conversation round refers to one interaction between the user and the system, and then encode each round of conversation using sentence-level BiGRU to obtain a vector representation of words in each round of conversation. Taking the ith round of dialog as an example, it is denoted as $h_i=[h_{i,1},...,h_{i,l}]$, which means the set of vector representations of l words in the ith round of dialog. The formula is shown below:

$$h_{i,j}^{forward}=GRU^{forward}(e(x_{ij}),h_{i,j-1}^{forward}) \tag{26}$$

$$h_{i,j}^{backward}=GRU^{backward}(e(x_{ij}),h_{i,j+1}^{backward}) \tag{27}$$

$$h_{i,j}=[h_{i,j}^{forward};h_{i,j}^{backward}] \tag{28}$$

where $GRU^{forward}$ and $GRU^{backward}$ are the same as described in the description of Eqs. 3-1, 3-2, x_{ij} denotes the j th word in the ith round of dialog, and $h_{i,j}$ denotes the vector representation of the j th word in the ith round of dialog obtained after passing through the bi-directional GRU network.

3) Based on the results of the sentence-level BiGRU encoder, a self-attention mechanism is then used to compute the integrated vector representations for each round of dialogues $ct=[ct_i,...ct_i,...]$. In this thesis, a simplified version of the self-attention mechanism computation method is implemented, and the computation formula is shown below for the ith round of dialogues as an example:

$$\alpha_{i,j}=softmax(W\cdot x_{ij}+b) \tag{29}$$

$$ct_i=\sum_{j=1}^{l}\alpha_{i,j}\cdot h_{i,j} \tag{30}$$

where W and b are model parameters and $\alpha_{i,j}$ denotes the self-attention weight of the j th word in the ith round of dialog.

4) After calculating the composite vector representation ct for each round of dialog, it is input to the dialog-level BiGRU encoder for computation, and its encoding result is used to calculate the composite vector representation of dialog history X using a simplified version of the self-attention mechanism CT. The computational formula is shown below:

$$s_i^{forward}=GRU^{forward}(ct_i,s_{i-1}^{forward}) \tag{31}$$

$$s_i^{backward}=GRU^{backward}(ct_i,s_{i+1}^{backward}) \tag{32}$$

$$s_i=[s_i^{forward};s_i^{backward}] \tag{33}$$

$$\alpha_i=softmax(W_0\cdot s_i+b_0) \tag{34}$$

$$CT=\sum_{i=1}^{T}\alpha_i\cdot s_i \tag{35}$$

where $GRU^{forward}$ and $GRU^{backward}$ are dialog-level BiGRU encoders with different internal parameters than sentence-level BiGRU, W_0 and b_0 are model parameters, and α_i is the self-attention weight for the ith round of dialog.

3.3.2. Knowledge encoder. Based on the candidate knowledge subsets filtered by the knowledge retrieval module as input, the knowledge encoder uses the static graph attention mechanism to compute the comprehensive vector representation kg of the knowledge subsets and the vector representation $K=[k_1,...k_i,...,k_s]$ of each knowledge triad, where $k_i =< head_i, rel_i, tail_i >, head_i, rel_i, tail_i$ represents the One-Hot representation of the head entity, relationship and tail entity of the i rd knowledge triad, respectively. The specific workflow is as follows:

1) First, the entities contained in each knowledge triad in the candidate knowledge subset are pre-coded, and in this thesis, the pre-trained TransE model is used. The computational formula is shown below:

$$(h_i, l_i, t_i) = TransE(head_i, rel_i, tail_i) \tag{36}$$

where h_i, l_i, t_i denotes the TransE precoded vectors for the head entity, relationship and tail entity, respectively.

2) The vector representation of each knowledge triad is computed using the multilayer perceptual machine model. The formula is as follows:

$$k_i = W \cdot [h_i; l_i; t_i] + b \tag{37}$$

where w and b are the model parameters, and the entity vectors in the triad are spliced for computational simplicity.3) Calculate the composite vector representation of the candidate knowledge subset using the static graph attention mechanism kg with the following formula:

$$\beta_j = (W_l \cdot k_j^l) \tanh (W_h \cdot k_j^h + W_t \cdot k_j^t) \tag{38}$$

$$\alpha_j = softmax(\exp(\beta_j)) \tag{39}$$

$$kg = \sum_{j=1}^{s} \alpha_j \cdot k_j \tag{40}$$

where W_l, W_h, W_t are all parameter matrices, k_j^l, k_j^h, k_j^t denotes the head entity vector, relationship vector and tail entity vector of the j rd knowledge triad multiple, respectively, and α_j denotes the self-attention weight of the j th triad.

3.3.3. Natural Language Generator. The natural language generator uses a memory network to dynamically interact with the GRU to generate system responses. Memory network is the memory, which is readable and writable, and its role is mainly to write each word vector $h=[h_1,...,h_i,...h_n]$ output from the dialog encoder and the output result $K=[k_1,...k_i,...,k_s]$ from the knowledge encoder into the memory, so that it can be read by the GRU when it dynamically generates words at each moment. The specific workflow is as follows:

1) First, at the time of external knowledge storage, the vector representation $K=[k_1,...k_i,...,k_s]$ of the knowledge triples in the candidate knowledge subset and the vector representation $h=[h_1,...,h_i,...h_n]$ of each of the dialog histories X are spliced together and written as inputs to the memory, which can be represented as $M=[h;K]=[M_1,\ldots,M_{n+s}]$ with a length of $n+s$.

2) Initialize the zero-moment hidden state s_0 of the GRU, i.e., $s_0=[kg;CT]$, using the composite vector representation of the candidate knowledge subset kg and the composite vector representation of the conversation history CT.

3) At each time step t of GRU decoding, GRU dynamically generates a query vector s_t for reading the knowledge memory. Before reading, the degree of correlation and joint representation between

the query vector s_t and each memory unit in M at time step t is first computed using the attention mechanism, as shown in the following formula:

$$\alpha_i^t = Attention(s_t, h_i) \tag{41}$$

$$c_t = \sum_{i=1}^{n} \alpha_i^t \cdot h_i \tag{42}$$

$$\beta_i^t = Attention(s_t, k_i) \tag{43}$$

$$g_t = \sum_{i=1}^{s} \beta_i^t \cdot k_i \tag{44}$$

where a_i^t denotes the degree of correlation between the query vector s_t and the vector representation h_l of the i rd word in the dialog history β_i^t denotes the degree of correlation between the query vector s_t and the vector representation k_i of the l th knowledge triad in the candidate knowledge subset, and in this dissertation, Attention is computed specifically by using Eq. 2-23.

4) Based on the query vector s_t and the joint representation c_t, g_t, access the knowledge memory and the lexicon to compute the knowledge distribution p_{ptr} as well as the lexicon distribution p_{vocab}, respectively, and utilize the gating mechanism to select the time step t when the word y_t is generated:

$$p_{ptr} = multihop([s_t; g_t], M) \tag{45}$$

$$p_{vocab} = MLP([s_t; c_t], V) \tag{46}$$

where multihop denotes the computational process of accessing the memory in multiple hops, [.] denotes splicing the vectors, V denotes the dictionary, and M is the memory. To compute the lexicon distribution, the query vector s_t is spliced with the joint representation c_t, and the multilayer perceptron model is used to compute the degree of correlation with each word in the lexicon to obtain the probability distribution on the lexicon.

In multi-hop reading of the knowledge memory, the query vector s_t is first spliced with the joint representation g_t, and after k hops it is computed to obtain q_k and p_{logit}, k is the hyperparameter defining the number of memory hops, q_k represents the result obtained after k hops of accessing the memory, and p_{logit} represents the degree of relevance of the description of each piece of knowledge on the k th hop. The calculation formula is shown below:

$$p^1 = softmax(M \cdot C^1 \cdot q^1) \tag{47}$$

$$o^1 = \sum_{i=1}^{n+s} M \cdot C^2 \cdot p^1 \tag{48}$$

$$q^2 = q^1 + o^1 \tag{49}$$

where the knowledge memory consists of a set of trainable parameter matrices $CK = [C^1, ...C^k]$, k i.e., the number of hops of the memory, C^1 is the parameter matrix of the first hop of the memory, and C^2 is the same. q^1 is the initial query vector, i.e., $[s_t; g_t]$, q^2 are the results obtained after one hop. The above formulas only show the computation of one hop, and multiple hops are realized by loops.

4. System interoperability testing

4.1. Simulation of Speech Interaction for English Learning

Simulation experiments are designed to verify the speech enhancement algorithm designed in this paper and its interaction quality.

In the experiment, the English network learning platform is set to have 20 microcells (Microcell), the number of users at the output end and the number of end-users in the number ratio of 8:2, the number of reference nodes for speech enhancement region selection are 10, 20, 30, 40 and 50, the number of rounds of synchronization cycle is 5, and the number of nodes for speech enhancement in the network learning platform are 100, 200, 300, 400 and 500, and the simulation parameters are set as shown in Table 1.

Table 1. Simulation parameter setting

Simulation parameter	Set value
Maximum packet size	128 Byte
Maximum network range	200* 400
Number of network nodes	200~800
Maximum perceptual radius	20 m
Maximum communication radius	60 m
Number of reference nodes	20~ 60
Number of beacon packages	6~ 18
Beacon interval	1 ms
Periodic number	1~ 12
Simulation time	12 h
Node initial energy	14 W·s
Sending end energy consumption	120 nW·s /bit
Receiving energy consumption	80 nW·s /bit

Based on the simulation environment and parameter settings described above, the voice interaction enhancement algorithm is tested, and the input voice signal is shown in Figure 6. The speech signal is interfered by a large amount of noise, and it is almost impossible to recognize the useful signal, resulting in poor quality of voice communication for English learning.

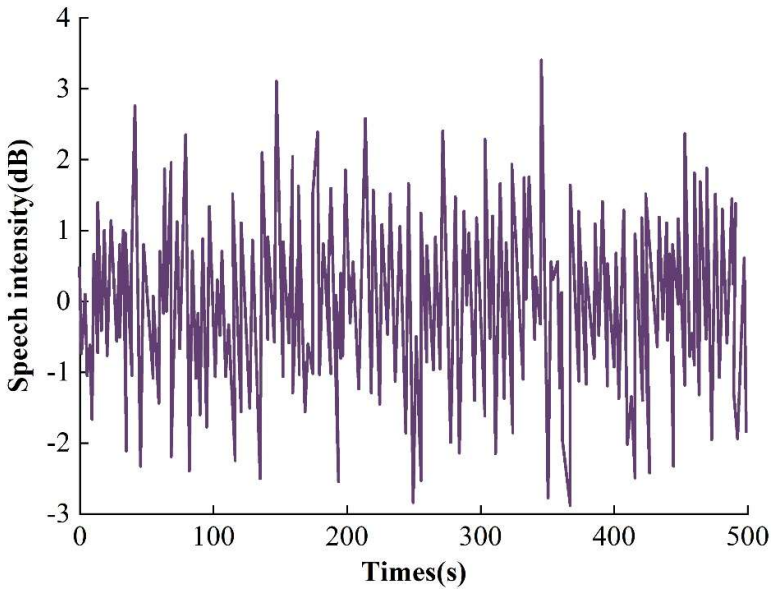
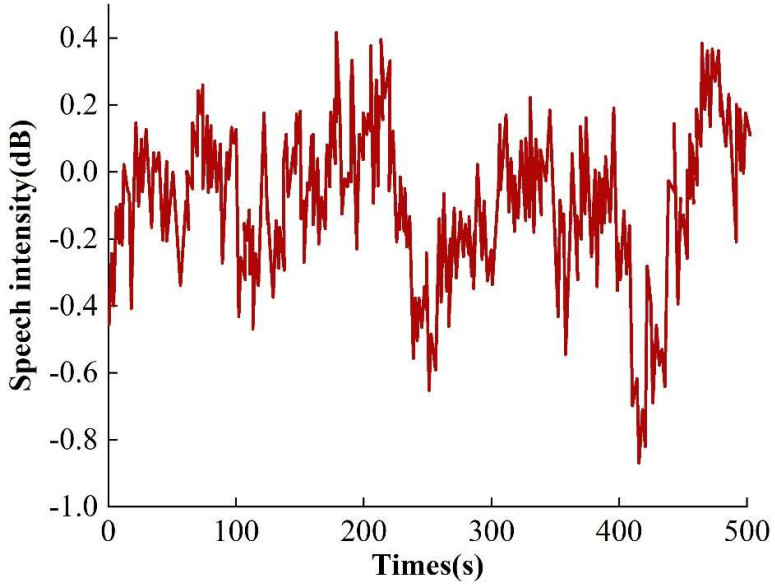


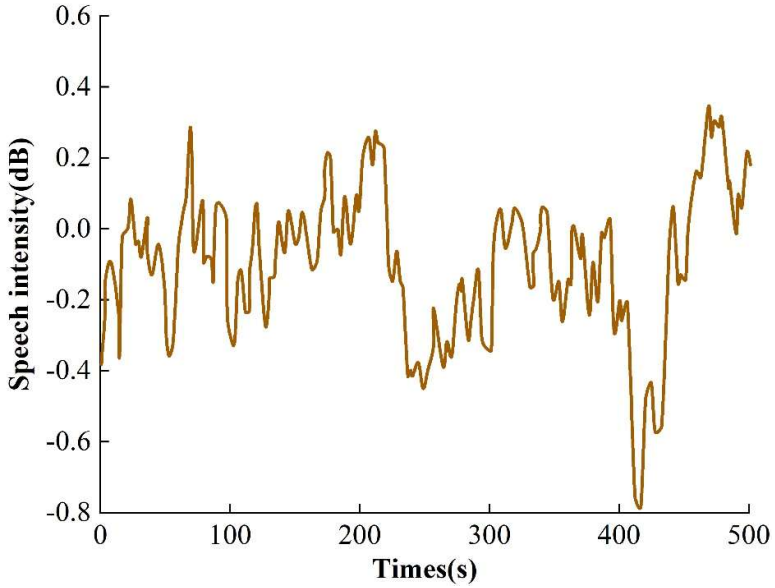
Fig. 6. Input voice signal

Taking the input signal of Fig. 6 as the test set, the present method is used to enhance the English learning speech communication signal with two thresholds, SNR = -10 d B and SNR = 0 dB, and the signal enhancement output is obtained as shown in Fig. 7. Where (a) (b) are the performance at -10 dB and 0 dB thresholds, respectively.

Comparing Fig. 6 and Fig. 7, it is learned that after the signal enhancement of English interactive learning speech communication by this method, the interference noise is effectively suppressed, the communication is stable, the output interference is reduced, and the signal enhancement is more effective with the increase of SNR.



(a)SNR=-10dB



(b)SNR=0dB

Fig. 7. Signal enhancement output

Taking network congestion as the evaluation index and using different methods for communication transmission, the comparison results are obtained as shown in Fig. 8. The average value of network congestion degree of this method is 0.073, which meets the reliable communication index and has a large reduction compared with the traditional method. In summary, the use of this method to achieve voice communication enhancement of English network learning platform improves the stability and reliability of communication and reduces interference, which reduces the degree of congestion and improves the data transmission rate.

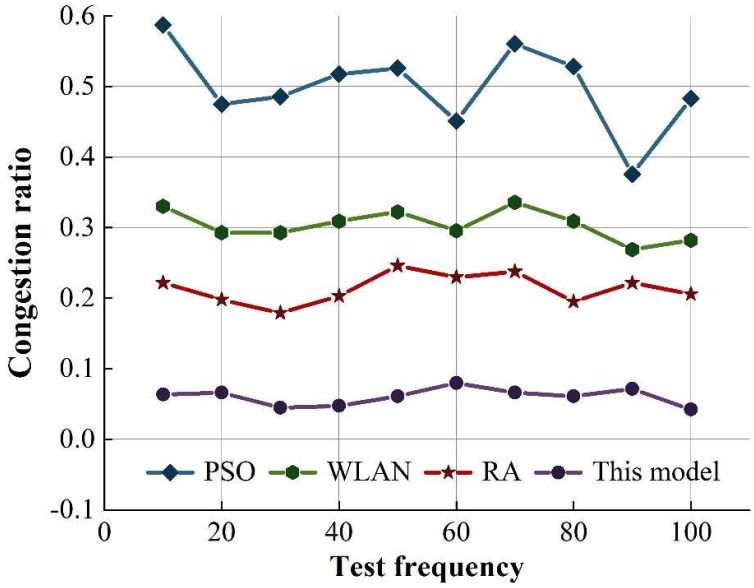


Fig. 8. Network congestion contrast

4.2. Results of the dialog state tracking experiment

Table 2 shows the performance comparison of the proposed model with three baseline models - TRADE, DSTReader and DSTQA. As shown in the table, the model outperforms all the baseline models in both slot accuracy and joint accuracy metrics, and improves 1.58% and 0.41% in both metrics, respectively. This indicates that the model proposed in this chapter is able to track the dialog state better.

Table 2. The performance contrast of dialog status tracking in the data set

Model	Joint Accuracy(%)	Slot Accuracy (%)
TRADE	48.53	96.86
DSTReader	41.13	96.24
DSTQA	51.56	97.38
This model	51.97	98.96

In order to investigate whether the modules of the model play a role, this paper designs an ablation experiment on three main modules, including natural language processing, knowledge retrieval, and natural language generation, and the experimental results are shown in Table 3. The joint accuracy of each module is below 50, which is lower than the overall system 51.78. It proves that the system structure is well designed.

Table 3. Ablation experiment results

Model	Joint Accuracy	Slot Accuracy
This model	51.78	98.35
Natural language processing	49.25	91.27
Knowledge retrieval	46.33	90.34
Voice enhancement	46.19	91.18
Recursive neural network	44.34	90.46

4.3. Experimental results of dialog response generation

The quality of dialog reply generation has a crucial impact on user experience and task completion, so this paper evaluates and explores the performance of the model in dialog reply generation. Table 4 shows the performance performance of dialog reply generation, from which it can be seen that the model proposed in this paper outperforms all the baseline models in terms of notification rate, success rate, BLEU value, and the final overall score in the dialog reply generation task of the dataset by improving at least 2.89, 3.02, 7.92, and 8.20 points, respectively.

Table 4. The performance comparison of the response generated by the dialogue

Model	Inform	Success	BLEU	Score
TRADE	74.16	59.62	16.85	84.38
DSTReader	82.28	79.45	12.45	93.51
DSTQA	83.05	68.16	26.44	99.45
This model	85.94	82.47	34.36	107.65

The inputs to the dialog reply generation module are the generated dialog state and the current user input. As mentioned previously in the introduction to the model, using conversation states instead of the complete conversation history can filter out information in the conversation history that is not relevant to completing the task and improve the model's ability to complete the task. To further explore the contribution of dialog states to model performance, this paper designs the following four comparison and ablation experiments: (1) Using dialog history instead of dialog states. (2) Using real dialog states instead of generated dialog states. (3) Input using only the current user input, neither dialog states nor dialog history. (4) Use the dialog state, dialog history and user's current input at the same time. The experimental results are shown in Table 5.

From the experimental results in the table, it can be seen that:

1) When using the dialog history instead of the dialog state, all four indicators are reduced to different degrees, and the final score is about 5 points lower than the original model. This may be due to the fact that on the one hand there is information in the dialog history that is not related to the completion of the task, and the longer the dialog, the more such information there is, which in turn causes greater interference in the selection of words when generating responses. On the other hand, modeling short texts is always easier than modeling long texts, and using dialog states instead of dialog history effectively shortens the length of the input sequence while retaining the information that is really helpful for completing the task, and thus is more helpful for performance improvement.

2) When using real dialog states instead of generated dialog states, there is a significant improvement in all the metrics. Since real dialog states can be regarded as the upper limit of the model's dialog tracking performance, this experiment shows that the model proposed in this paper is greatly affected by the performance of dialog state tracking. In other words, as the conversation progresses, the

conversation history becomes longer and longer, and the ratio of the information contained in the conversation state to the information contained in the current user input becomes larger and larger, so for a conversation scenario with a high average number of conversation rounds such as multi-tasking hybrid human-machine conversation, the ability to accurately generate the conversation state is critical to the final ability to complete the task.

3) When neither dialog state nor dialog history is input, but only the user's current utterance is input, the model basically loses the ability to memorize the history information, so the model performs very poorly, which also confirms the importance of the dialog state to the model's ability to generate replies in another way.

4) When dialog state, dialog history and user's current input are input at the same time, the performance of the model is slightly lower than that of the case where only dialog state and user's current input (i.e., the original setting) are input. This is due to the fact that the dialog state already contains the core information in the dialog history that helps to complete the task, and adding the dialog history at this point will result in more distributional noise on the input sequence, which affects the choice of words when generating replies.

Table 5. Comparison and ablation of input information

Model	Inform	Success	BLEU	Score
The model proposed in this article	85.26	80.38	24.08	107.42
Enter dialogue history	82.43	75.16	23.36	102.48
Enter the real dialogue state	93.78	84.27	25.97	114.09
Only enter the user current statement	42.15	35.12	15.24	54.32
Dialogue status + dialogue history	84.19	79.28	23.61	106.11

Similar to the dialog state tracking task, this section also designs ablation experiments for the dialog reply generation task, the results of which are shown in Table 6. It can be seen that the main modules of the model also have a more significant impact on the dialog reply generation task, and the trend of performance degradation is also close to that during the dialog state tracking experiment. This is because the dialog states fed into the dialog reply generation module are generated by the previous module, and thus the generated dialog states have a significant impact on the performance of reply generation, as evidenced by the experimental results when using real dialog states in Table 5 from another perspective.

Table 6. The ablation experiment of the main module

Model	Inform	Success	BLEU	Score
This model	85.74	80.24	25.36	107.48
Natural language processing	82.16	78.35	22.15	103.26
Knowledge retrieval	79.25	79.16	19.35	101.79
Voice enhancement	80.39	77.37	19.74	102.23
Recursive neural network	81.65	78.28	20.58	101.34

4.4. Subjective assessment

There is a lack of effective evaluation standards at home and abroad for the expression of naturalness in human-computer interaction dialog. Here, a subjective evaluation method was used to evaluate the multimodal HCI system of this study, in which 50 evaluators participated in the experiential evaluation

of the interactive system for English learning based on ChatGPT and deep reinforcement learning. These 50 reviewers were compared using the HCI system proposed in this study for the following three modalities:

1) Speech-only interaction, in which the user only inputs speech and the system uses speakers without avatars to output synthesized speech.

2) The user interacts with the system in a multimodal way, but the system does not process the user's emotional information, while the digital avatar adopts a randomized way to process the expression and gesture animation output at the output end.

3) The user interacts with the emotion expression system using the natural language processing human-computer interaction proposed in this paper.

The reviewer will give a mean opinion score (MOS) to these animations according to the following criteria:

(1) Natural, behaves like a natural human-to-human dialog process.

(2) Fairly natural, behaves close to a natural human-human conversation, but not perfect.

(3) Moderate, average performance, slightly stilted form of dialog.

(4) Unnatural, poor experience of natural human-human dialog.

(5) Completely unnatural.

The results of averaging the scores given by the raters are shown in Figure 9, where the scores of the three groups are 2.156, 2.794, and 3.766, respectively, where it can be seen that the human-computer dialog patterns generated by the first two methods are more unnatural. The human-computer interaction English learning method proposed in this paper can make the human-computer interaction experience more smooth, make the interaction between the virtual person and the user seem more natural and lively, obviously improve the naturalness of human-computer conversation and the expressiveness of the digital virtual person, and improve the user's interactive experience of English learning.

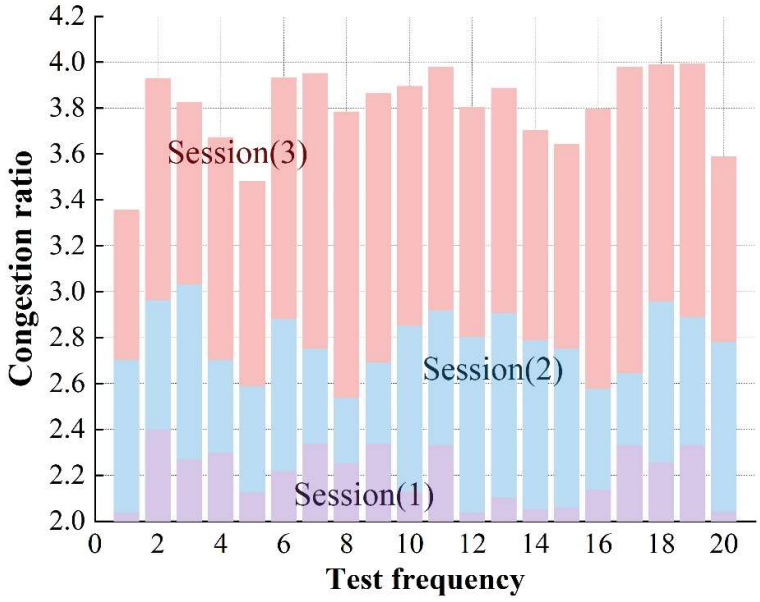


Fig. 9. The use of MOS scoring in different interaction modes

5. Conclusion

In this paper, we design a human-computer dialog interaction system for English learning scenarios based on Deep Reinforcement Learning, ChatGPT and Artificial Intelligence interaction technology. The speech enhancement method based on deep reinforcement learning can improve the quality of

voice communication, reduce interference and congestion rate. And the mean value of congestion of this network is 0.073, which satisfies the reliable communication index, and there exists a reduction of at least 0.1 or more compared with the baseline method. The system in this paper also outperforms the baseline model in both the conversation state tracking and conversation reply generation tasks, especially in the conversation reply generation task where it has a lead of about 7.92 points in the BLEU score compared to the second place baseline model. In addition, the analysis results also show that the model is able to capture the implicitly mentioned slot values when generating dialog states, and is able to generate systematic replies that help to complete the English learning task with fluent utterances.

References

[1] Liu, S. (2024). Machine Translation-Based Language Modeling Enables Multi-Scenario Applications of English. Machine Translation, 15(9).

[2] Tupe, N. (2015). Multimedia Scenario Based Learning Programme for Enhancing the English Language Efficiency among Primary School Students. International Journal of Instruction, 8(2), 125-138.

[3] Song, D., Yang, E., Guo, G., Shen, L., Jiang, L., & Wang, X. (2024). Multi-scenario and multi-task aware feature interaction for recommendation system. ACM Transactions on Knowledge Discovery from Data, 18(6), 1-20.

[4] Xu, S., Li, L., Yao, Y., Chen, Z., Wu, H., Lu, Q., & Tong, H. (2023, February). MUSENET: Multi-scenario learning for repeat-aware personalized recommendation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining (pp. 517-525).

[5] Mbatchou, G. M., Bouchet, F., & Carron, T. (2018). Multi-scenario modelling of learning. In Innovations and Interdisciplinary Solutions for Underserved Areas: Second International Conference, InterSol 2018, Kigali, Rwanda, March 24–25, 2018, Proceedings 2 (pp. 199-211). Springer International Publishing.

[6] Han, M., & Niu, S. (2021). Application of virtual scenario teaching in spoken English teaching. International Journal of Emerging Technologies in Learning (iJET), 16(18), 129-142.

[7] Escudero, P., Smit, E. A., & Mulak, K. E. (2022). Explaining L2 lexical learning in multiple scenarios: cross-situational word learning in L1 Mandarin L2 English speakers. Brain Sciences, 12(12), 1618.

[8] Mahbub, I. S. P., & Hadina, H. (2021). A systematic overview of issues for developing efl learners'oral english communication skills. Journal of Language and Education, 7(1 (25)), 229-240.

[9] Feng, H. (2024, January). Research on Construction of Scenario-based English Learning Based on Personalized Recommendation and Recurrent Neural Networks. In 2024 IEEE 7th Eurasian Conference on Educational Innovation (ECEI) (pp. 321-323). IEEE.

[10] Yi, Q., Wu, L., Tang, J., Zeng, Y., & Song, Z. (2024). Hybrid contrastive multi-scenario learning for multi-task sequential-dependence recommendation. Neural Networks, 106953.

[11] Ning, Y. (2022). A new model of multiple intelligence for teaching English informatics in the IoT scenario. Computational Intelligence and Neuroscience, 2022(1), 5642284.

[12] Elbourhamy, D. M., Najmi, A. H., & Elfeky, A. I. M. (2023). Students' performance in interactive environments: an intelligent model. PeerJ Computer Science, 9, e1348.

[13] Wang, S. (2021, January). Innovative thinking and practice of mobile interaction design teaching in artificial intelligence era. In Proceedings of the 2021 12th International Conference on E-Education, E-Business, E-Management, and E-Learning (pp. 215-219).

[14] Mavrikis, M., & Holmes, W. (2019). Intelligent learning environments: Design, usage and analytics for

future schools. Shaping future schools with digital technology: An international handbook, 57-73.

[15] Gros, B. (2016). The design of smart educational environments. Smart learning environments, 3, 1-11.

[16] Cheung, S. K., Kwok, L. F., Phusavat, K., & Yang, H. H. (2021). Shaping the future learning environments with smart elements: challenges and opportunities. International Journal of Educational Technology in Higher Education, 18, 1-9.

[17] Wang, Y., Eysink, T. H., Qu, Z., Yang, Z., Shan, H., Zhang, N., ... & Wang, Y. (2022). Interactive response system to promote active learning in intelligent learning environments. Journal of Educational Computing Research, 60(7), 1867-1891.

[18] Herder, E., Sosnovsky, S., & Dimitrova, V. (2017). Adaptive intelligent learning environments. Technology enhanced learning: Research themes, 109-114.

[19] Zhang, H., Wang, Z., Zong, S., Wu, H., Jiang, R., Cui, Y., ... & Luo, H. (2024). Impact of intelligent learning environments on perception and presence of hearing-impaired college students. Educational Technology & Society, 27(4), 352-374.

[20] Marougkas, A., Troussas, C., Krouska, A., & Sgouropoulou, C. (2021). A framework for personalized fully immersive virtual reality learning environments with gamified design in education. In Novelties in Intelligent Digital Systems (pp. 95-104). IOS Press.

[21] Kinshuk, Chen, N. S., Cheng, I. L., & Chew, S. W. (2016). Evolution is not enough: Revolutionizing current learning environments to smart learning environments. International Journal of Artificial Intelligence in Education, 26, 561-581.

[22] Ukenova, A., & Bekmanova, G. (2023). A review of intelligent interactive learning methods. Frontiers in Computer Science, 5, 1141649.

[23] Xiao Zeng,Shiyun Xu & Mingjiang Wang. (2024). A time-frequency fusion model for multi-channel speech enhancement.EURASIP Journal on Audio, Speech, and Music Processing(1),47-47.

[24] Amitabha Govande,Raju Attada & Krishna Kumar Shukla. (2024). Predicting PM2.5 levels over Indian metropolitan cities using Recurrent Neural Networks. Earth Science Informatics(1),1-1.

[25] Xingyu Zhang,Yanshan Wang,Yun Jiang,Charissa B Pacella & Wenbin Zhang. (2024). Integrating structured and unstructured data for predicting emergency severity: an association and predictive study using transformer-based natural language processing models. .BMC medical informatics and decision making(1),372.