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

Combinatorial Press

Context-Aware Translation Accuracy Improvement Strategies Based on Deep Reinforcement Learning in English Translation

Li Shi[1], Xiaohong Sun[1,] ✉

[1] *Shijiazhuang Information Engineering Vocational College, Shijiazhuang, Hebei, 050000, China*

ABSTRACT

The field of machine translation has made significant progress in recent years, but how to improve translation accuracy and context consistency is still an urgent challenge. In this paper, a context-aware translation accuracy improvement strategy based on deep reinforcement learning is proposed for English translation. Based on CNNs neural machine translation model, the multi-intelligence deterministic deep policy gradient algorithm is utilized to combine the output of the translation model with the human evaluation index (BLEU), and the reward function is constructed to guide the model learning. In addition, in order to enhance the context-awareness of the model, the study introduces a context encoder in the deep reinforcement learning framework to capture sentence-level contextual information and incorporate it into the translation process. The experimental results show that the optimized model has better training performance, with 40 epochs of iterations, the Loss converges to 0.135 up and down, and its English translation F1 value is 94.95%. And as the number of encoder layers rises, the number of semantic high-level features increases. The N-GRR difference between the generated translation and the standard translation of the model in this paper is the smallest, and the over-translation phenomenon is less. The number of out-of-set word interference is more than 6, and the BLEU value of this paper's model is improved by 17.89% to 55.55% compared with the comparison model. And the algorithm has good translation performance, with METEOR scores of 0.562~0.803 on different topics. The research results fully verify the effectiveness of deep reinforcement learning based on deep reinforcement learning to improve the accuracy of English machine translation.

Keywords: Deep Reinforcement Learning, Machine Translation Model, Policy Gradient Algorithm, Context Encoder, English

✉ Corresponding author.
E-mail address: Susanna8010@163.com (X. Sun).
Received 15 April 2024; Revised 05 July 2024; Accepted 25 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-203
© 2025 The Author(s). Published by Combinatorial Press. This is an open access article under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the continuous development of science and technology, artificial intelligence plays an increasingly important role in various fields. In the field of language translation, AI translation system has gradually replaced the traditional machine translation technology and become an essential tool in people's life and work [1-4]. However, to improve the translation quality of AI translation system, one of the key issues is how to realize the accurate understanding of context [5-6].

Context refers to textual information in a specific context, and in language communication, understanding context is the key to ensure accurate translation. Traditional machine translation systems are mainly based on statistical models and lack in-depth understanding of context, which leads to errors in translating complex sentences or involving semantic ambiguity [7-10]. In contrast, AI translation systems can better understand the contextual information through deep learning and other technologies, thus improving the quality of translation. Deep learning is currently a popular technology in the field of artificial intelligence research, and its application in context understanding has made significant progress [11-14]. The AI translation system based on deep reinforcement learning can be trained with a large number of corpora to learn the correlation between contexts. By utilizing the contextual information in the context, the system is able to better deal with semantic ambiguity, word sense disambiguation, and other problems to improve translation accuracy [15-18].

Based on the English translation task, this study fuses deep reinforcement learning with context encoder to optimize the CNNs neural machine translation model for context-aware translation accuracy. The probability distribution of each word is obtained by forward propagation of the neural machine translation model. The reward value for the occurrence of words at each position is estimated after combining the average value of BLEU of the sentence composed of each word according to the policy gradient algorithm. Using the gradient descent algorithm, the model parameters are updated and the optimized model is finally obtained. In addition, experimental analyses of model training, translation performance, and translation quality are conducted to further verify the effectiveness of the strategy in capturing contextual information and generating more natural translation results.

2. Deep Reinforcement Learning-Based Strategies for Improving English Translation Accuracy

2.1. Basis of research

2.1.1. Research on English Machine Translation Methods. According to the different cognitive viewpoints, machine translation research methods can be categorized into rule-based methods, statistical-based methods and example-based methods, etc. Rule-based methods use rules to depict linguistic knowledge such as the internal structural laws of natural languages, and use the rules to guide computer translation. However, since rules are not easy to depict those empirical and small-grained knowledge in natural languages, rule-based methods lack flexibility in dealing with complex linguistic phenomena. In order to overcome the shortcomings of rule-based approaches, statistical-based approaches and instance-based approaches, also called them corpus-based approaches, have emerged. They use large corpora in the hope of obtaining the needed linguistic knowledge directly from large-scale real texts stored in the corpus. Statistical-based methods use statistical methods to calculate the parameters of a probabilistic model of language translation from a corpus and then use the model to translate. The current Chinese-English machine translation research methods show a diversified trend.

2.1.2. Deficiencies in Chinese-English conversion:

1) Generation of English Verb Tenses. English verbs have tense grammatical categories, and their tenses need to be determined in order to make corresponding morphological changes when generating English. Chinese verbs do not have tense grammatical categories, which cannot provide a basis for the generation of English verb tenses.

2) Determination of Omitted Components. At present, most of the Chinese-English machine translation systems adopt the single-sentence analysis mode, and the quality of the translation decreases when the subject of the Chinese sentence is omitted and affects the determination of the subject of the English translated sentence.

3) English Article Generation. The definite article in the English translation can describe the definite information of the noun phrase, but the Chinese noun phrase does not have obvious formal markers to indicate the definite meaning, and the analysis in the scope of a single sentence only cannot determine the definite meaning of the Chinese noun phrase according to the context, which affects the generation of articles.

4) Singular and Plural Noun Phrases. English nouns have singular and plural morphological changes, so when converting between Chinese and English, it is necessary to know whether a Chinese noun phrase is singular or plural. Chinese nouns do not have morphological changes, and when there is no clear word in the sentence to indicate the quantity, most of the current machine translation systems just use the singular form to convert, which makes it difficult to ensure the accuracy of the process.

2.2. Key technologies

2.2.1. Deep reinforcement learning. Figure 1 shows the process of visualizing environment interaction in reinforcement learning. a standard reinforcement learning model [19] consists of an intelligent body. The intelligent body observes the state information s_t of its environment at moment t and, based on this state information, selects the best action a_t to execute according to the current decision. Accordingly, the current environment transitions to the next new state s_{t+1} as a result of the execution of the intelligent body's action, and at the same time feeds back an immediate reward value R_t associated with its action.

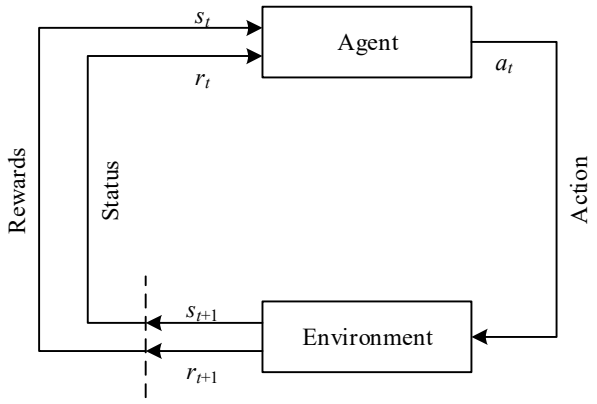

Fig. 1. Enhanced learning of central boundary interaction visualization

The above process is represented using Markov Decision Process (MDP) [20]. The MDP problem can be represented by a quaternion $\langle S, A, P, R \rangle$ containing four basic elements, where $S = \{s_0, s_1, s_2, \cdots, s_t, s_{t+1}, \cdots\}$ is the set of states, $A = \{a_0, a_1, a_2, \cdots, a_t, a_{t+1}, \cdots\}$ is the set of actions of the intelligent, and P is the state transfer probability matrix, which denotes the probability that the

system will choose action a_t when in state s_t. $R = r(s,a)$ is the payoff reward function, which measures how good it is to take a certain action at a certain state.

Intelligence generates a large amount of data by continuously interacting with the environment, and uses this data to learn and optimize its own strategy for choosing actions π. The goal of training is to find the best strategy π that maximizes the long-term cumulative reward value. The cumulative return value is affected by the return value at each future moment, introducing $\gamma \in [0,1)$ as a discount factor, and γ as a coefficient used to determine the magnitude of the effect that a possible future state has on the present state. The long-term return value R_t is shown in Eq:

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{k=1}^{\infty} \gamma^k r_{t+k+1} \tag{1}$$

To address the limitations of reinforcement learning, an innovative algorithm that combines the advantages of reinforcement learning and deep learning is introduced, namely the deep reinforcement learning algorithm. The algorithm fully utilizes the superior decision-making capabilities of reinforcement learning and combines the unique perceptual advantages of neural networks for automatic extraction of features from high-dimensional data as well as the fitting of nonlinear equations in deep learning. Neural networks are able to quickly extract complex features from high-dimensional inputs, replacing the manual feature extraction process in traditional reinforcement learning. In addition, the use of neural networks to fit the policy function and the abandonment of linear equations to characterize the value equations can achieve a more accurate and flexible regulation of the intelligent body's actions. Deep reinforcement learning is mainly divided into two categories: value-based deep reinforcement learning algorithms and deep reinforcement learning algorithms based on policy gradient.

2.2.2. Multi-Intelligent Deterministic Depth Policy Gradient Algorithm. The Deterministic Deep Policy Gradient (DDPG) algorithm [21], combines the DPG algorithm with the DQN algorithm. For better exploration, the algorithm obtains a new exploration strategy μ' by adding noise N to the deterministic strategy $\mu_\theta(s)$:

$$\mu'(s) = \mu_\theta(s) + \mathrm{N} \tag{2}$$

In addition, DDPG's updates for both actor and criterion parameters are soft updates. Instead of using the parameters of the new strategy directly, soft updates introduce a parameter τ □ 1 when the parameters are updated and use that parameter for the update operation:

$$\theta' \leftarrow \tau\theta + (1-\tau)\theta' \tag{3}$$

The Multi-Intelligent Deterministic Deep Policy Gradient Algorithm [22] (MADDPG) extends the DDPG algorithm, where the criterion in MADDPG is a centered criterion, and centered means that the criterion has access to the relevant data of all the intelligences. Each intelligence learns a value function for its own behavior:

$$Q_i^{\vec{\mu}}\left(\vec{o}, a_1, \ldots, a_N\right) \tag{4}$$

It can be seen that the inputs to Q use the data associated with all the other actuators, where $a_1, \ldots, a_N$ they are the actions performed by the N intelligences. The $Q_i^{\vec{\mu}}$ functions of each intelligent body are trained and learned independently, and the structure of their rewards may appear arbitrary, including the fact that they may compete for rewards. Each intelligent body i also possesses an actor that is used to perform policy exploration as well as policy parameter θ_i updating. In this paper, we will use this algorithm in combination with a context encoder to improve the accuracy of the machine translation model, and the architecture of the MADDPG algorithm is shown in Fig. 2.

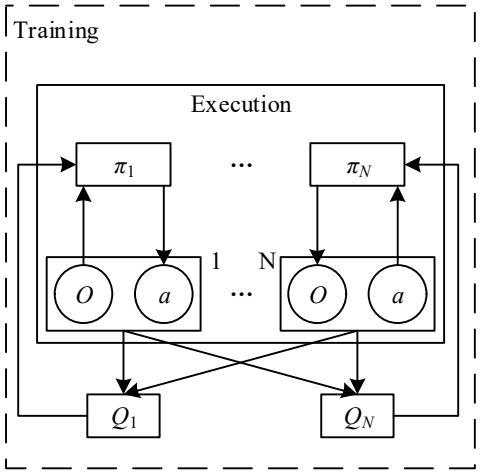

Fig. 2. Architecture diagram of MADDPG

2.2.3. Context Encoder. In this paper, we use a Transformer-based word-level encoder [23] to encode each translated utterance and thus represent the semantic information of that translated utterance. For each translated utterance in the translation history $u_j^i = \{w_1^j, w_2^j, ..., w_{|u_j|}^j\}$, this paper uses the encoder part of Transformer as the main model of the word-level encoder. Two types of embedding representations are taken as inputs, i.e., identifier embedding, and positional embedding. The identifier embedding is the word vector generated by dimensionality reduction with a fully-connected linear layer after converting identifiers to uniquely hot coding, denoted as E_{token}, and the positional embedding encodes the sequentiality of the input sequence. In this paper, we use the trainable sine-cosine absolute position embedding as the position embedding, denoted as E_{pos}, which is computed as follows:

$$\begin{aligned} P(x) &= \left[..., \sin(w_i x), \cos(w_i x), ...\right]^T; w_i \in \square \\ E_{pos} &= \left[P(1), P(2), ..., P(N)\right] \end{aligned} \tag{5}$$

where $P(x)$ is the absolute position vector of the xnd bit of the input sequence, w_i is the trainable parameter, and N is the length of the input sequence. In addition, before embedding the translated utterances into the representation, the corresponding special identifiers are added to identify the beginning and end of the input text. If the input translated utterance contains multiple single sentences, <s> and <\s> are used as special identifiers to denote the beginning and end of the sentence, respectively. In addition, this paper uses <eou> as a special identifier to denote the end of each translated utterance, which is added at the end of each translated utterance. Eventually, the representation of each translated utterance u_j^i input to the input translation sample is as follows:

$$u_j^i = \left\{< s >, w_1^j, w_1^j, ..., < \backslash s >, < eou >\right\} \tag{6}$$

Input u_j^i into each of the three embedding layers and perform the embedding representation transformation, denoted E_w:

$$E_w = E_{token} + E_{pos} \tag{7}$$

2.2.4. CNNs Syntax-Supervised Neural Machine Translation Models. The model basis of the optimization strategy in this paper is CNNs neural machine translation model. Figure 3 shows the overall structure of CNNs neural machine translation model. The neural machine translation model is trained using the training dataset, the preprocessed data are input into the model, and the grammar

parsing decoder and translation decoder are trained at the same time, the grammar parsing decoder takes the sequence of grammatical blocks as the label for supervised training, and the translation decoder takes the corresponding Chinese sentences as the supervised training labels, which completes the initial training of the model.

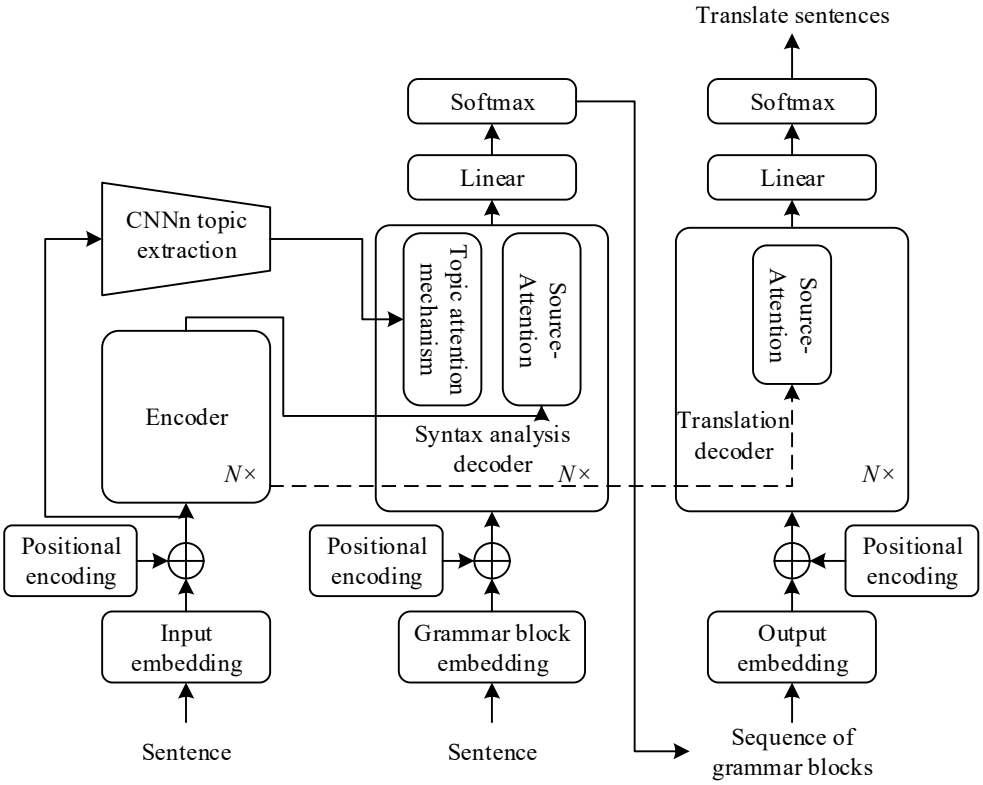

Fig. 3. The overall structure of the CNNs neural machine translation model

First stage decoding: the syntactic parsing decoder integrating CNNs sentence context topic attention module autoregressively predicts the syntactic parsing block sequences, the English source sentence attention i.e., the output of the encoder is denoted by s, the block identifiers are denoted by $c_1,...,c_n$, and n is the length of the syntactic block sequences, as in Eq:

$$p(c_1,...,c_n|s)=\prod_{i=1}^{n} p(c_i|c_1,...,c_{i-1},s) \tag{8}$$

Second stage decoding: a single non-autoregressive step is applied to generate the Chinese target sentence by decomposing the target sequence probability into the following form, where T is the target sentence length, the English source sentence note i.e., the output of the encoder is denoted by s, the grammar block identifier is denoted by $c_1,...,c_n$, and n is the length of the current sequence of grammatical blocks, as in Eq:

$$p(y_1,...,y_T|s)=\prod_{i=1}^{T} p(y_i|c_1,...,c_n,s) \tag{9}$$

2.3. Deep Reinforcement Learning Optimization of English Neural Machine Translation Research

2.3.1. Accuracy optimization strategies. In this paper, deep reinforcement learning is used to optimize and fine-tune the parameters of the model with the goal of sentence-level indicator BLEU

(Bilingual Evaluation Understudy) value, and this method can alleviate some of the over-turning and under-turning problems.

The deep reinforcement learning algorithm MADDPG is applied to the non-autoregressive CNNs neural machine translation model, and a context encoder is introduced into it to enhance the context-awareness of the model, defining the optimized model as the Deep Reinforcement Learning Set Context Encoded Neural Machine Translation Model (DRLCENMT). A significant advantage is that each token in non-autoregressive translation is independent from each other, and there is no need to base the next translation token on the previous translation token, which refers to the words in the sequence. Using this independence, the expected loss function can be expressed as follows: using the reward rewards obtained from sampling as weights, the probability distribution generated independently for each token as a strategy function, and the loss function of all positions is summed up to take a negative number. And the reward reward of each token is obtained by averaging the BLEU values calculated by sampling the whole sentence N times after fixing the current token. The specific optimization process is described below.

2.3.2. Reward estimation. The probabilistic model for non-autoregressive translation is shown in Eq:

$$P(Y \mid X,\theta) = \prod_{i=1}^{T} p\left(y_i \mid X,\theta\right) \tag{10}$$

where X is the input to the translation model, Y is the target utterance of the predicted output, T is the length of the target sentence, θ is the neural network parameters, i is the position i in the sentence, y_i is the predicted word at the i position in the sentence, and $P(\ )$ denotes the probability function.

The gradient of the expected loss of the described reinforcement learning paradigm is shown in Eq:

$$\nabla_\theta L_\theta = -\sum_{Y} \nabla_\theta \prod_{i=1}^{T} p\left(y_i \mid X,\theta\right) \cdot r(Y) \tag{11}$$

In Eq. $r(\ )$ denotes the reward computation function whose input is the whole sentence Y and the output is the BLEU value of this sentence. ∇_θ denotes the gradient of the neural network parameters θ, and Y denotes the sentence obtained by reinforcement learning sampling.

For non-autoregressive models, the above equation can be simplified as:

$$\nabla_\theta L_\theta = -\sum_{i=1}^{T} \sum_{y_i} \nabla_\theta p\left(y_i \mid X,\theta\right) \cdot r\left(y_i\right) \tag{12}$$

where $r\left(y_i\right)$ is the expected reward when the vocabulary y_i is fixed, is computed as follows:

$$r\left(y_i\right) = \mathrm{E}_{y_{i,-1}} \mathrm{E}_{y_{i+1},T} r(Y) \tag{13}$$

To estimate the REWARD reward, start with the first word of a sentence, fix this position, fix the word in this position, and sample all the other words from the probability n time, and compute the average of these n sentence REWARDS as the value of the reward for this strategy for the occurrence of this word in this position. $p\left(y_t \mid X,\theta\right)$ is the strategy. Then replace the word in this position and repeat the sampling to evaluate the REWARD. After everything in this position has been evaluated, $t+1$ look at the next position and continue the same sampling and estimation.

2.3.3. Expectation Gradient Estimation. Expectation Estimation Gradient $\nabla_\theta L_\theta$ The flow of the algorithm is as follows:

Input: output probability distribution $p(\cdot \mid X, \theta)$, traversal count k, number of samples n.

Output: estimate t position of $\nabla_\theta L_\theta$ according to Eq.

1) $T_k =$ {words that rank in the top $top-k$ in probability $p(\cdot \mid X, \theta)$}.

2) $\nabla_\theta L_\theta = 0$, $\overline{p} = p$, $P_k = 0$.

3) for y_t in T_k do.

4) Sample n times to estimate $r(y_t)$.

5) $\nabla_\theta L_\theta -= \nabla_\theta p(y_t \mid X, \theta) \cdot r(y_t)$.

6) $\overline{p}(y_t \mid X, \theta) = 0$.

7) $P_k += p(y_t \mid X, \theta)$.

8) Regularize $\overline{p}(y_t \mid X, \theta)$.

9) Sample y_t from $\tilde{p}(y_t \mid X, \theta)$.

10) Use Algorithm 1 to sample n times to estimate $r(y_t)$.

11) $\nabla_\theta L_\theta -= (1 - P_k) \cdot \nabla_\theta \log\left(p(y_t \mid X, \theta)\right) \cdot r(y_t)$.

12) return $\nabla_\theta L_\theta$.

Finally the time complexity of the whole algorithm is $O(T_k \cdot n)$. The space complexity is $S(n) = O(1)$. The gradient $\nabla_\theta L_\theta$ is estimated using the REINFORCE formula as in Eq:

$$\nabla_\theta L_\theta = -\sum_{t=1}^{T} \mathrm{E}_{y_t}\left[\nabla_\theta \log\left(p(y_t \mid X, \theta)\right) \cdot r(y_t)\right] \tag{14}$$

The above corresponds to the gradient estimation method obtained by sampling the target word y_t, and the expected gradient is estimated using the log-probability gradient of y_t obtained by weighting the rewards r. Although the gradient estimate is unbiased, its variance is still large.

Unbiased estimation can be established by traversing the top-k words and then estimating the remaining words by sampling once, as in Eq:

$$\begin{aligned} \nabla_\theta L_\theta = -\sum_{t=1}^{T} \Bigg(& \sum_{y_t \in T_k} \nabla_\theta p(y_t \mid X, \theta) \cdot r(y_t) + (1 - P_k) \\ & \cdot \mathrm{E}_{y_r \sim \overline{p}}\left[\nabla_\theta \log\left(p(y_t \mid X, \theta)\right) \cdot r(y_t)\right] \Bigg) \end{aligned} \tag{15}$$

where $p(\cdot \mid X, \theta)$ is the output probability distribution of the target word generated by the decoder at position t.

T_k is the set of probabilities for a series of top-k target words.

P_k is the sum of the probabilities in T_k.

$\overline{p}$ is the normalized probability distribution after removing the word probabilities in T_k.

The algorithm takes as input the output probability distribution $p(\cdot \mid X, \theta)$, the traversal count k and the number of samples n, and outputs the gradient estimation of step t. The gradient estimation process of step t is first divided into two parts: traversal and sampling.

After obtaining the gradient $\nabla_\theta L_\theta$, the neural network parameters θ are updated based on the following equation to fine-tune to obtain the new neural network parameters θ_{new}:

$$\theta_{new} = \theta + \alpha \nabla_\theta L_\theta \tag{16}$$

where α is the learning rate. After that, the loop is iterated again to perform Monte Carlo sampling, estimate the REWARD, estimate the expected gradient, and update the model parameters.

2.4. Experimental design

2.4.1. Data sets. This paper uses two parts of the dataset: the first part is the collected parallel corpus, which contains 250,000 items. The second part is the pseudo-parallel corpus generated by DBIS method, which contains 1 million items.The 250,000 parallel corpus and 1 million pseudo-parallel corpus are categorized into several topics: literature, business, law, science and technology, medicine, news and so on. All 1.25 million corpora are divided into training set and validation set in 7:3 ratio for model training and accuracy improvement test experiments.

2.4.2. Experimental setup. The optimizer used for the study is Adam optimizer using Gelu activation function with a learning rate of 0.0001. The learning rate tuning strategy is linear preheating and inverse square root decay. All experiments were trained using hardware of Tesla V100s 32GB GPUs compute card. PyTorch version 1.12.0 was used, Python version number 3.7, and PyCharm was used for programming.

3. Context-awareness accuracy improvement validation experiment and result analysis

3.1. Optimizing Model Training and Language Feature Detection

In this section, in order to improve the performance of the neural machine translation model for its context-aware translation accuracy in English translation, it is optimized using deep reinforcement learning, and the parallel corpus collected above is used for training. The model is set to iterate for 200 epochs, and the training loss variation curve of the machine translation model optimized based on deep reinforcement learning is shown in Figure 4. In the figure, in the process of model training on the dataset, the optimized model in this paper achieved rapid convergence in the 30th iteration, and its performance gradually stabilized in the subsequent iteration rounds, especially after the 40th round. The model loss curve shows slight fluctuations in the convergence stage, and the loss value oscillates around 0.135, which reflects the high accuracy and stability of the model training. The optimized model's loss reduction rate within the first 40 epochs of the training phase is significantly faster than the subsequent phases, which further verifies the model's efficiency in the initial training phase. For the performance evaluation of the validation set, the optimized model also demonstrated superior stability. This feature is mainly attributed to the introduction of the reward mechanism in the neural machine translation model, which effectively accelerates the training process, a feature that makes the optimized model in this paper more reliable and robust in real-world applications in English.

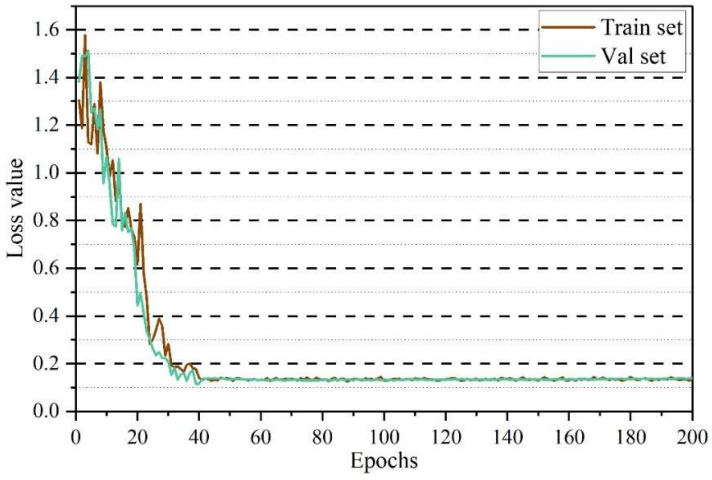

Fig. 4. Optimization machine translation model training loss curve

In order to verify the effectiveness of this paper's neural machine translation model optimized based on deep reinforcement learning on English translation, this section chooses to compare with the Rule-Based Machine Translation Model (RBMT), Statistical Machine Translation (SMT), Low-Resource Machine Translation (LRMT), Multi-Modal Machine Translation (MMT), and Neural Machine Translation (NMT), the five typical translation models on the above mentioned test corpus respectively, and A comparative evaluation of translation performance was done. The commonly used performance evaluation indexes are used for evaluation, i.e., accuracy rate, recall rate and F1 value, usually the higher F1 value indicates the better model performance.

The results of the English translation comparison experiments with different models are shown in Figure 5. From the figure, it can be seen that in the English translation task in the test dataset, the neural network model optimized with deep reinforcement learning in this paper achieves the best results compared to the other five mainstream translation models. The accuracy, recall, and F1 value reached 95.32%, 94.59%, and 94.95%, respectively, proving its usability in the English translation task. Among them, the unoptimized NMT model has limitations in feature extraction, which makes it difficult to obtain deep contextual semantic information, and thus the unoptimized NMT model has a 17.77% lower F1 value than this paper's model. It shows that the model in this paper has good translation performance and generalization ability.

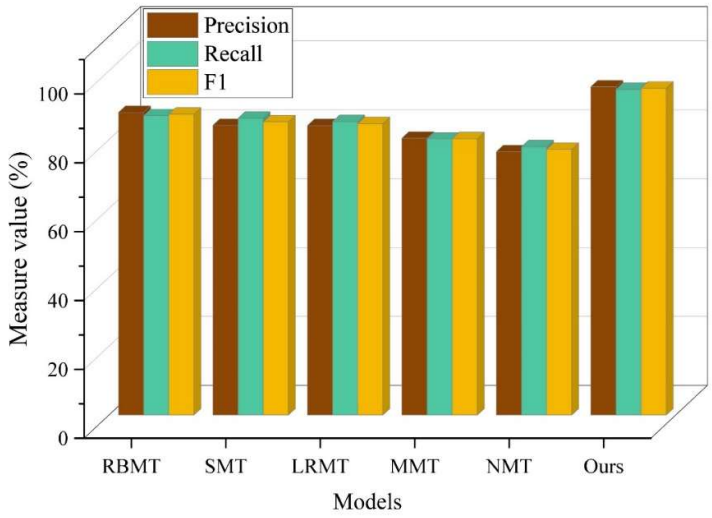

Fig. 5. Different models of English translation comparison experimental results

Linguistic feature detection tasks can be viewed as classification problems aimed at investigating the ability to recover specific linguistic features from encoder hidden states. In this paper, we use 10 specific detection tasks, categorized into three main types:

1) Shallow features: sentence length (SeLe), word content (WC).

2) Syntactic features: tree depth (TrDep), top constituent (ToCo), binary semantic shift (BShit).

3) Semantic features: tense (Tense), subject number (SubN), object number (ObjN), semantic singular form (SoMo), collocation inversion (CoIn).

In this section, linguistic feature detection experiments are conducted for different hidden states of the encoder, i.e., different levels of source language representations, as a way to analyze the expressive behavior of different linguistic features in the source language representations from a linguistic point of view. Detection experiments are performed on the Transformer model of the word-level encoder. That is, the encoder is 6 layers, and then the detection experiments are conducted for the word vector layer, the 1st layer, the 3rd layer and the 6th layer, and the detection results of the hidden state language features of the word-level Transformer encoder are shown in Figure 6. In order to better show the change trend of language features at different levels, the

experimental results of the word vector layer (Embedding layer) are set as the baseline in this section, and the detection experimental results of other layers are scaled.

From the figure, it can be seen that as the number of encoder layers rises, the shallow features (e.g., lexical features, etc.) encoded in the hidden state decreases consequently, while the high-level features (syntactic and semantic features) become more and more numerous. Thus, it can be seen that the bottom layer of the encoder contains more shallow features and fewer syntactic and semantic features. Furthermore, the results of the linguistic feature detection experiments show how capturing useful linguistic features from source language sentences is crucial for the quality of the source language representation.

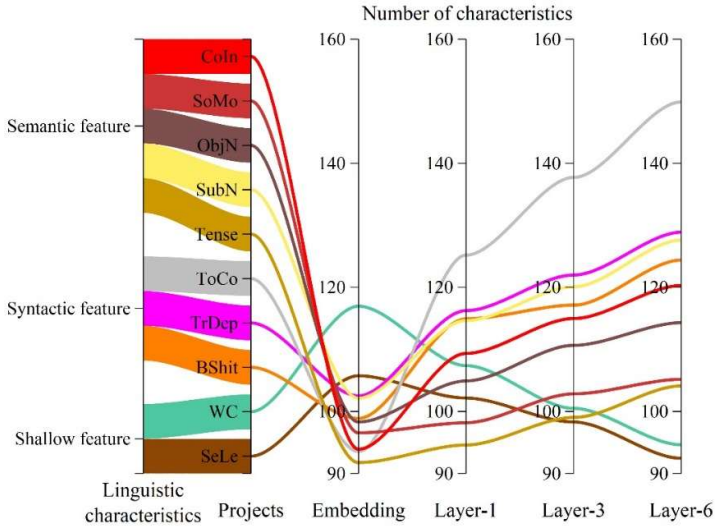

Fig. 6. The word level transformer is hidden from the state language characteristics

3.2. Comparative Analysis of Over-Translated Texts

The experiments in this section test the models' ability to handle the over-translation problem and conduct quantitative analysis experiments on the English translation task in the dataset. In this section, the N-generic repetition rate metric (N-GRR) is used to measure the severity of the over-translation problem for different models. The metric is computed in the form of the proportion of N-meta-word repetitions in a sentence:

$$N-GRR=\frac{1}{CR}\sum_{c=1}^{C}\sum_{r=1}^{R}\frac{\left\|N-grams_{c,r}\right\|-\left\|u(N-grams_{c,r})\right\|}{\left\|N-grams_{c,r}\right\|} \tag{17}$$

where $\left\|N-grams_{c,r}\right\|$ denotes the total number of N-metric words in the r rd translation corresponding to the c nd source sentence in the corpus, while $\left\|u(N-grams_{c,r})\right\|$ denotes the size of the set after removing duplicate N-metric words. A lower value of $N-GRR$ indicates a less severe case of over-translation. In the test data of this section, there are a total of C =7816 sentences. In the standard translation, the number of translations r = 5, while in the machine translated translation, the number of translations r = 1.

The over-translations of the different models are shown in Figure 7. It can be seen that compared with the standard translation, the neural machine translation model (NMT) is more prone to over-translation phenomenon, and its N-GRR value is higher than that of the standard translation by 4.56~6.05. This is mainly because neural machine translation does not have a relevant mechanism to ensure that each source word can be translated and can only be translated once like statistical machine translation. The optimized model in this paper has the smallest N-GRR difference with the standard translation, with a maximum difference of only 0.72. The use of deep reinforcement

learning and context encoder to introduce contextual sentence information at the target end alleviates this problem to a large extent. It further illustrates the adjunctive nature of deep reinforcement learning and context encoder to the translation task.

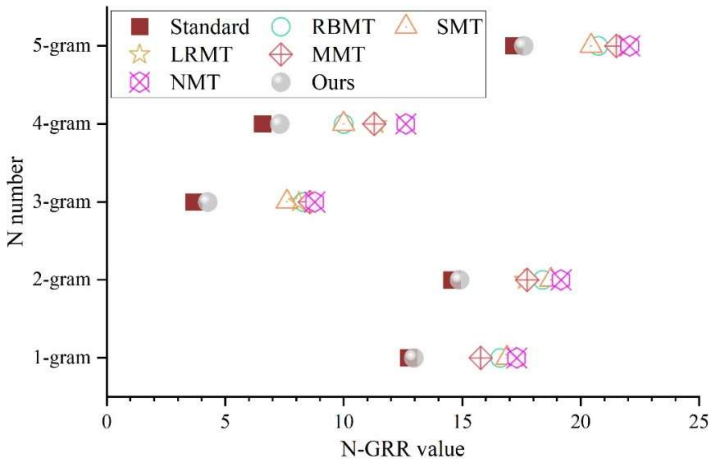

Fig. 7. Over translation of different models

3.3. Out-of-set word impact and similarity mean difference assessment

In order to further verify the ability of the proposed method to mitigate the negative effects of out-of-set words, the entire test set is grouped according to the number of out-of-set words contained, e.g., "5" means that all the test set sentences in the group contain only 5 out-of-set words, and then the corresponding BLEU values for each group are calculated:

$$BLEU = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log P_n\right) \tag{18}$$

where n is the n of the n-gram and w_n is the weight for different n-grams. Firstly, the number of times each word appears inside each candidate word is counted, and then the number of times each word appears in the reference translations is counted, Max denotes the maximum value in several reference translations, and Min denotes the minimum value of the candidate translations and Max.

The translation evaluation results of the source language sentence sets with different numbers of out-of-set words are shown in Fig. 8. As can be seen from the figure, when the number of out-of-set words is 0, the optimization model of this paper has a BLEU value of 30.02, which exceeds that of all the other compared methods, while the performance of the other compared methods is almost the same, with a BLEU of about 28.1. This is probably due to the fact that the context-awareness strategy adopted in this paper can strengthen the representation of in-set words for improving translation prediction. As the number of out-of-set words increases, the optimized model in this paper also outperforms the comparison models. In particular, when there are more than 6, the BLEU values of this paper's model are improved by 17.89% to 55.55% compared with the comparison methods. That is, the deep reinforcement learning and context-aware strategy used in this paper performs relatively better than other comparison methods.

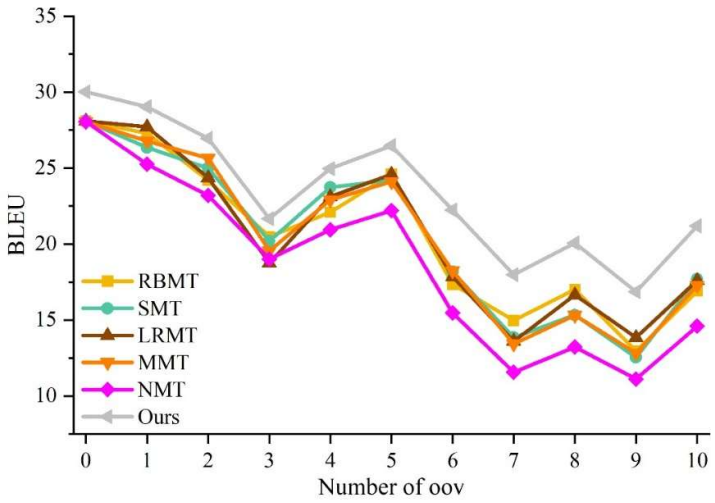

Fig. 8. Different number of integrated translation evaluation results

In order to have a more intuitive understanding of why the model improves the performance of translation systematicity, some experimental results are analyzed from two perspectives. Namely, ASG(ss) denotes the average difference in similarity between positive and negative example phrase pairs at the source end, and ASG(ts) denotes the average difference in similarity between positive and negative example phrase pairs at the target end. These two metrics are used to verify whether the model is able to distinguish between positive and negative examples of phrase pairs better, and a larger average difference indicates that the model performs well in capturing semantic differences.

First, parallel phrase pairs are extracted from the database to construct negative examples. Then calculate the average difference between the similarity of positive and negative samples in both directions for phrases learned by different models. Taking the source side as an example, the calculation is as follows:

$$\frac{1}{N}\sum(sim(f,e)-sim(f,e')) \tag{19}$$

where $sim(f,e)$ denotes the similarity between phrase f and phrase e, e' is the negative example of phrase e, and N is the total number of phrases.

The result of the similarity difference between phrase pairs of positive and negative examples in the dataset is shown in Fig. 9. From the figure, it can be seen that in most cases, the optimization model in this paper has the largest similarity difference in both ASG(ss) and ASG(ts) directions. Taking ASG(ss) as an example, the optimization model in this paper measures 0.231, while the maximum value of the comparison model (MMT) is only 0.184. From this experimental result, it can be concluded that the neural machine translation model of GNNs, which incorporates the deep reinforcement learning and the context encoder, is able to translate phrase pairs more accurately.

The results show that after adding contextual encoding, the optimized model further improves the translation quality compared with the comparison model. For example, the contrast model does not translate "chaos" correctly, but rather "confusion". The reason for this is that "confusion" appears more frequently in the dataset than "chaos" (165:81), and the comparison model does not have contextual information, so it does not give correct translation results. In contrast, in documents on law-related topics, "chaos" is more likely to be translated as "chaos". By adding contextual encoding, the optimization model in this paper correctly translates "chaos" into "chaos".

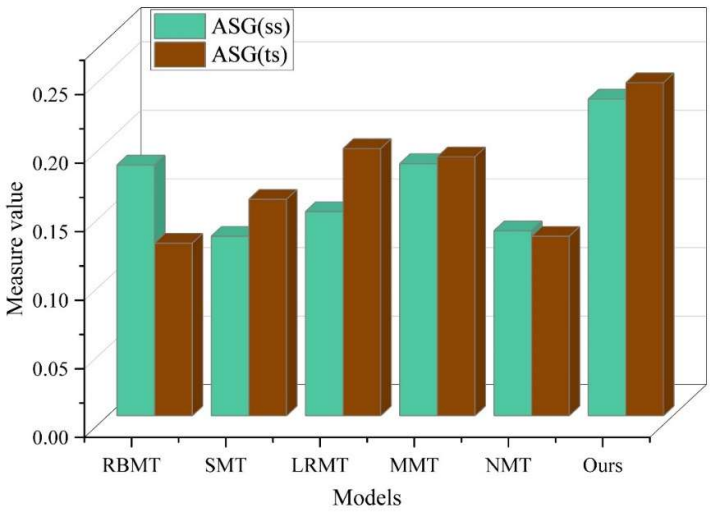

Fig. 9. The results of the similarity difference of the plus and negative cases

3.4. BLEU and METEOR measurements for different subject contexts

In this section of the experiments, BLEU and METEOR are chosen as the metrics for automatic evaluation.METEO (Metric for Evaluation of Translation with Explicit Ordering) is a metric to measure the quality of translations based on the average of the accuracy and recall.METEOR at the sentence or paragraph level correlates well with human assertions and is calculated as follows:

$$F_{mean} = \frac{10PR}{R+9P} \tag{20}$$

$$METEOR = F_{mean}(1-\frac{0.5(c)^3}{m}) \tag{21}$$

The translation BLEU values of the optimization model in different topics of the dataset are shown in Figure 10. The data in the figure shows that based on the comparison model, the translation results of the optimized model in this paper are improved. The optimized neural machine translation model has BLEU values of 29.94, 38.89, 32.37, 38.48, 33.51, and 32.29 for the topics of Literature, Business, Law, Technology, Medicine, and News, respectively.The BLEU values of the optimized model are higher than those of the pure neural machine translation model (NMT) by 7.24 to 9.45 BLEU values. These results emphasize the effectiveness of the optimization model in this paper in the English translation task, showing its potential in improving translation accuracy. Such experimental comparisons help to more comprehensively evaluate the performance of different models in different English subject translation tasks.

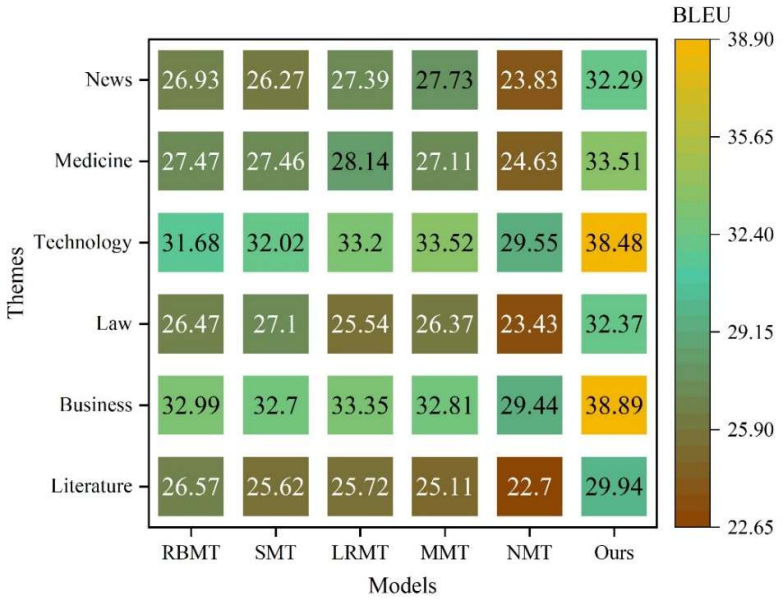

Fig. 10. Optimize the translation of BLEU values in different topics in the data set

In addition, Fig. 11 shows the results of METEO comparison during the translation process of different models.METEOR combines accuracy and recall. The METEOR scores of this paper's optimization model on different topics in the translation task range from 0.562 to 0.803, which is a significant improvement over the comparison models, especially the NMT model. This suggests that the introduction of reinforcement learning algorithms and context encoders helps to improve a more comprehensive translation quality metric.

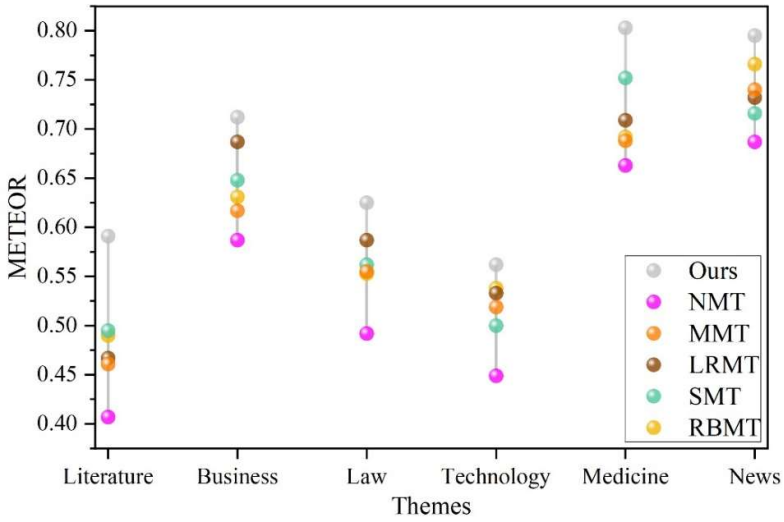

Fig. 11. METEOR contrast results in different model translation

These experimental results provide clues for improvement, and the robustness of the experimental results is verified after further experiments and adjustments. It is shown that the DRLCENMT model proposed in this paper for contextual information integration is more conducive to translation information mining and can better improve context-aware translation accuracy and language fluency.

4. Conclusion

Aiming at the shortcomings of traditional machine translation in English-Chinese conversion, the article designs a strategy to improve the accuracy of context-aware translation. The strategy is centered on the deep reinforcement learning-based policy gradient algorithm, optimized with the addition of a context encoder to the CNNs neural machine translation model. The algorithm is utilized to repeatedly sample words at different positions and calculate the average value of the reward of the sentence in which the word is located. Monte Carlo sampling is then performed to estimate the expected gradient and update the model parameters. The context encoder encodes each translated utterance to achieve an accurate representation of the contextual semantic information of that translated utterance. The model shows good training effect on the training and validation sets, and its accuracy, recall and F1 value in the translation process are 95.32%, 94.59% and 94.95%, respectively. In addition, the maximum difference between the generated translation and the standard translation on the N-GRR over translation evaluation index is only 0.72, which is much lower than the comparison model. Regardless of the out-of-set word interference, the BLEU value of this paper's model is higher than that of the comparison model, and its ASG(ss) measure reaches 0.231, which has better performance in capturing semantic differences. Meanwhile, the optimization model has the largest BLEU value and METEOR score among the translated translations of different topics, showing its excellent effect in English translation.

References

[1] Mohamed, Y. A., Khanan, A., Bashir, M., Mohamed, A. H. H., Adiel, M. A., & Elsadig, M. A. (2024). The impact of artificial intelligence on language translation: a review. Ieee Access, 12, 25553-25579.

[2] Kolhar, M., & Alameen, A. (2021). Artificial Intelligence Based Language Translation Platform. Intelligent Automation & Soft Computing, 28(1).

[3] Khasawneh, M. A. S., & Al-Amrat, M. G. R. (2023). Evaluating the Role of Artificial Intelligence in Advancing Translation Studies: Insights from Experts. Migration Letters, 20, 932-943.

[4] Das, A. K. (2018). Translation and Artificial Intelligence: Where are we heading. International Journal of Translation, 30(1), 72-101.

[5] Li, R., Nawi, A. M., & Kang, M. S. (2023). Human-machine Translation Model based on Artificial Intelligence Translation. Journal for ReAttach Therapy and Developmental Diversities, 6(10s), 1122-1129.

[6] Naveen, P., & Trojovský, P. (2024). Overview and challenges of machine translation for contextually appropriate translations. Iscience, 27(10).

[7] McDonald, S. V. (2022). Accuracy, readability, and acceptability in translation. Applied Translation, 16(2), 1-9.

[8] Akramovna, T. U. (2024). TRANSLATION EQUIVALENCE: THE KEY TO ACCURATE CROSS-CULTURAL COMMUNICATION. Ethiopian International Journal of Multidisciplinary Research, 11(12), 172-175.

[9] Ashengo, Y. A., Aga, R. T., & Abebe, S. L. (2021). Context based machine translation with recurrent neural network for English–Amharic translation. Machine translation, 35(1), 19-36.

[10] CHENG, Y., WANG, R., CHEN, J., CHAO, Y., Maimaitili, A., & ZHANG, H. (2023). Context-Based AI Translation From a Globalization Perspective: A Case Study of ChatGPT. Sino-US English Teaching, 20(9), 370-380.

[11] Li, S., & Huang, Y. (2024). BAS-ALSTM: analyzing the efficiency of artificial intelligence-based English translation system. Journal of Ambient Intelligence and Humanized Computing, 15(1), 765-777.

[12] Ahmed, N. (2023). A review of existing Machine Translation Approaches, their Challenges and Evaluation Metrics. Pakistan Journal of Engineering, Technology & Science, 11(1), 29-44.

[13] Panaro, J. (2020). Fine-Tuning Sign Language Translation Systems Through Deep Reinforcement Learning. Rochester Institute of Technology.

[14] Jia, Y., Yang, X., & Cui, Q. (2024). Research on the Role of Artificial Intelligence in the Core of Intelligent Translation Systems. Procedia Computer Science, 243, 585-592.

[15] Guo, X. (2022). Optimization of English machine translation by deep neural network under artificial intelligence. Computational Intelligence and Neuroscience, 2022(1), 2003411.

[16] Dong, H., Dong, H., Ding, Z., Zhang, S., & Chang, T. (2020). Deep Reinforcement Learning. Singapore: Springer Singapore.

[17] Shahmerdanova, R. (2025). Artificial Intelligence in Translation: Challenges and Opportunities. Acta Globalis Humanitatis et Linguarum, 2(1), 62-70.

[18] He, J. (2022, November). Research on Machine Translation (MT) System Based on Deep Reinforcement Learning. In 2022 3rd International Conference on Computer Science and Management Technology (ICCSMT) (pp. 487-490). IEEE.

[19] Zhipeng Zhou,Wen Zhuo,Jianqiang Cui,Haiying Luan,Yudi Chen & Dong Lin. (2025). Developing a deep reinforcement learning model for safety risk prediction at subway construction sites. Reliability Engineering and System Safety(PB),110885-110885.

[20] J.J.J. Pey,S.M. Bhagya P. Samarakoon,M.A. Viraj J. Muthugala & Mohan Rajesh Elara. (2025). A Decentralized Partially Observable Markov Decision Process for complete coverage onboard multiple shape changing reconfigurable robots. Expert Systems With Applications126565-126565.

[21] Xiaomei Zhang,Fan Yang,Qiwen Jin,Ping Lou & Jiwei Hu. (2023). Path Planning Algorithm for Dual-Arm Robot Based on Depth Deterministic Gradient Strategy Algorithm. Mathematics(20), 4392-.

[22] Qiang Gao,Shuhao Li,Yuehui Ji,Junjie Liu & Yu Song. (2025). Scalable path planning algorithm for multi-Unmanned Surface Vehicles based on Multi-Agent Deep Deterministic Policy Gradient.Ocean Engineering120243-120243.

[23] Kévin Cédric Guyard,Jonathan Bertolaccini,Stéphane Montavon & Michel Deriaz. (2025). A Transformer Encoder Approach for Localization Reconstruction During GPS Outages from an IMU and GPS-Based Sensor. Sensors(2),522-522.