International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 1470-1485

https://doi.org/10.70517/ijhsa464133

A Progressive Adaptation Architecture for Deep Learning Translation Models in Engineering English Education

Ailing Zhang[1,*] and Yongchang Zhang[2]
[1] School of General Education, Jiangsu Vocational Institute of Architectural Technology, Xuzhou, Jiangsu, 221116, China
[2] School of Information and Electrical Engineering, Jiangsu Vocational Institute of Architectural Technology, Xuzhou, Jiangsu, 221116, China
Corresponding authors: (e-mail: Alley324@163.com).

Abstract Neural machine translation models have made significant progress in general-purpose domains, but there are still many challenges for the models to solve translation tasks in specialized domains with low resources, especially in how to make full use of terminology information. The study firstly expresses the fundamentals of neural machine translation from three perspectives: encoder-decoder framework, text feature representation and decoding search, respectively. Then, it introduces BLEU, the evaluation index at the core of neural machine translation, and proposes engineering domain adaptive techniques and data enhancement methods. Finally, improvements are made on the outside of the translation model, and a neural machine translation model fusing the underlying information is proposed. Aiming at the problem that the English-Chinese translation model in the field of electrical engineering does not fully utilize terminological information, the terminological vocabulary is taken as a priori knowledge. Through experimental validation, the method can significantly improve the quality of machine translation and show more satisfactory translation results in low-resource languages as well, and the method proposed in this paper improves the BLEU values by 1.25-1.82 on average on different datasets. In addition, the model proposed in this paper outperforms the translation performance of the traditional Transformer model in the Chinese-English translation task. In summary according to the BLEU value of the model, the training time rise reaches the enhancement and achieves the balance of translation quality and time cost.

Keywords neural machine translation, deep learning, terminology information, domain adaptive technology, engineering English education

I. Introduction

Engineering education is the main channel to provide engineering technology and enterprise management talents for national construction and social development. China enters the critical period of industrial restructuring and economic transformation and upgrading in urgent need of a group of applied, compound and international engineering talents [1]-[4]. The engineering English course is a public elective or compulsory course commonly offered in applied engineering colleges and universities, and its teaching objective is to cultivate applied engineering talents who have mastered certain language foundation and skills, possessed professional knowledge and literacy, and possessed the ability to combine English language skills with professional abilities, and able to satisfy the needs of China's economic development and international exchange market [5]-[8].

However, engineering English is more objective and involves a large number of specialized vocabulary and terminology, some words have specialized word meanings in the field of engineering in addition to their basic word meanings, and some long sentences or structural logic have their own characteristics, which are quite different from ordinary English education, and the translation of engineering English is more complicated [9]-[12]. Deep learning, as a machine learning method, can be used to solve complex problems by training neural networks, and has achieved remarkable results in many fields, including speech recognition, image recognition and natural language processing [13]-[16]. In engineering English teaching, deep learning can be applied to speech recognition, text classification, machine translation, automatic summarization and personalized learning. By utilizing the advantages and technical means of deep learning, students' reading ability and learning effect can be improved [17]-[20].

In order to improve the translation effect of engineering English education and promote the development of engineering industry, this paper uses the collected electrical engineering corpus to train the neural machine translation model with Transformer as the baseline model. It introduces the basic principles of neural machine translation as well as BLEU, the core evaluation index of neural machine translation, from three perspectives: encoder-decoder framework, text feature representation and decoding search. On this basis, an additional underlying feature extraction layer is added for extracting the shallow syntactic information of the source language,

and its output of the shallow syntactic information of the source language is fused with the deep semantic information of the source language output from the last layer of the encoder unit using residual concatenation. Finally, a comparison experiment is designed to introduce the experimental setup, and the experimental results are analyzed to prove the translation effect of the translation model that fuses the underlying features.

II. Deep Learning based Neural Machine Translation Models

II. A. Deep learning based modeling framework

II. A. 1) Fully connected networks

A fully connected network is a typical hybrid mathematical model, a network structure consisting of alternating stacked combinations of linear and nonlinear models. The arithmetic process of the model is as follows:

$$h_1 = W_1 x + b_1 \tag{1}$$

$$z = \sigma(h_1) \tag{2}$$

$$h_2 = W_2 z + b_2 \tag{3}$$

$$\hat{y} = soft\max(h_2) \tag{4}$$

where, x represents the input, which is generally the features of the sample, W_1, b_1, W_2, b_2 is the parameters to be learned in different layers, h_1, h_2 represents the result of linear computation in each layer, and $z, \hat{y}$ represents the intermediate result and the final predicted output after nonlinear transformation, respectively.

Fully connected neural networks are the basis of deep learning models, other network models such as CNN, RNN, etc. are developed based on fully connected networks, the reason why other network models appear is because fully connected neural networks with the increase in the number of layers, the model will be very complex, both in terms of computation and storage, it will be very tricky, so it gave birth to other networks.

II. A. 2) Attention model

In order to solve the problems of the ordinary encoder-decoder framework, scholars have proposed the attention model. The attention model is an improvement work based on the ordinary encoder-decoder framework. Specifically, the fixed-length encoding vectors in the ordinary encoder-decoder framework are converted into an attention model that adjusts the parameters according to the results of the output. In the attention model based encoder-decoder framework, each output result at the decoder side is defined as a probabilistic model represented as follows:

$$P(y_i \mid y_1, y_2, ..., y_{i-1}, X) = g(y_{i-1}, s_i, c_i) \tag{5}$$

where X denotes the source language sentence information, y denotes the generated target translation sentence information, and $g(\cdot)$ is a nonlinear transformation function. In order to facilitate the explanation of the model, a specific model, such as the encoder-decoder framework based on RNN-RNN, is used as an example to develop the explanation. In the above equation, s_i denotes the hidden state at moment i in the RNN model at the decoder side, and c_i is the context vector obtained by the encoder using the fused attention model. The specific calculations are as follows:

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \tag{6}$$

$$c_i = \sum_{j=1}^{T_r} a_{ij} h_j \tag{7}$$

In the formula, the subscript i represents the i nd word or character in the encoder; h_j the hidden state vector of the j th word or character in the RNN model at the encoder side; α_{ij} represents the weight between the j th word at the encoder side and the i th word at the decoder side, which can be interpreted as the influence of the j th word at the source side on the i th word at the target side, and the α_i utilizes the Softmax model in shallow machine learning. The calculation is as follows:

$$a_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} \tag{8}$$

The Softmax model converts the weights into probability values such that the sum of the weights generated by the different words at the source side for the words at the target side is 1. e_{ij} is the alignment model, which is used to measure the degree of alignment or influence of the j nd word at the encoder side, on the i rd word at the decoder side, and more specifically, e_{ij} is computed as follows:

$$e_{ij} = a(s_{i-1}, h_j) \tag{9}$$

Equation (9) is a formal calculation of the alignment model, $a(\cdot)$ called the attention function. According to $a(\cdot)$ the attention model can be categorized into additive attention model, multiplicative attention model, self-attention model and so on. The computational expression of additive attention model is as follows:

$$a(s_{i-1}, h_j) = v_a^T \tanh(W_a s_{i-1} + U_a h_j) \tag{10}$$

where W_a, U_a, v_a denotes the weight matrix, the advantage of the additive attention model is that the computation can be minimized by calculating in advance.

The multiplicative attention model utilizes the characteristics of the matrix, and the computational complexity is similar to that of the additive attention model. The computational expression of the multiplicative attention model is as follows:

$$a(s_{i-1}, h_j) = s_{i-1}^T W_a h_j \tag{11}$$

The multiplicative attention model is easy to compute and has high storage performance because it takes advantage of matrices. But for high dimensional computation, the performance is not as good as additive attention model.

Self-attention model is a fusion of additive attention model and softmax model, which is calculated as follows:

$$A = soft\max(v_a \tanh(W_a H^T)) \tag{12}$$

$$H = (h_1, h_2, ..., h_T) \tag{13}$$

where H is the hidden state vector corresponding to the input sequence, W_a is the weight matrix, v_a is the parameter vector, and the result of the calculation A represents the attention matrix. The self-attention model can perform feature extraction by focusing on itself, and does not require additional information.

II. B.How Neural Machine Translation Works

Language modeling is the foundation of natural language processing, and its role is to express natural language into a mathematical form that can be processed by computers. The first use of neural networks to construct a language model was made in 2003, which was the beginning of the application of neural networks in the field of natural language processing [21]. Let V be a list of words, for sentence $Y = \{y_1, y_2, ..., y_T\}, y_i \in V$, the language model is approximately equal to a series of parameters θ such that the formula (14) is satisfied.

$$P(Y;\theta) = P(y_1, y_2, ..., y_T; \theta) \tag{14}$$

The essence of neural machine translation, on the other hand, is the autoregressive language model, which represents the decoding of the output words one by one while decoding the generated target language, and the previous decoding results will be used as inputs for decoding the current words. The mathematical model so constructed is shown in Eq. (15), where conditional probability modeling is carried out for a given source language sentence $X = \{x_1, x_2, ..., x_T\}$ condition, T' representing a source language sentence sequence of length T', for a possible generated target language sentence $Y = \{y_1, y_2, ..., y_T\}$, T representing a target language sentence sequence of length T, where $y_0 \sim t-1 = y_0, y_1, ..., y_{t-1}$.

$$P(Y \mid X;\theta) = \prod_{t=1}^{T+1} P(y_t \mid y_{0\sim t-1}, X;\theta) \tag{15}$$

At initial state $t = 0$, y_0 represents the beginning of the sequence and y_{n+1} represents the end of the sequence. Usually, for autoregressive machine translation models, the maximum likelihood estimation is used for training, and the loss function is the cross-entropy loss, and the training process of neural machine translation is the process of maximizing Eq. (16), where N represents that there are N parallel corpus pairs.

$$L^{ML}(\theta)=\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n+1}\log p(y_t^n \mid y_{0\sim t-1}^n, x_{1\sim T}^n;\theta) \tag{16}$$

In practice, improved gradient descent algorithms such as the Adadelta optimizer or the Adam optimizer are used for faster and better training results.

II. B. 1) Encoder-Decoder

Natural language processing tasks are generally sequences, such as a sentence or a piece of speech, and neural machine translation is a “sequence-to-sequence” task model. In 2014, research scholars first proposed the “sequence-to-sequence” model, and then improved on this basis to determine the core “encoder-decoder” structure of the current neural machine translation, and its corresponding structure is shown in Figure 1.

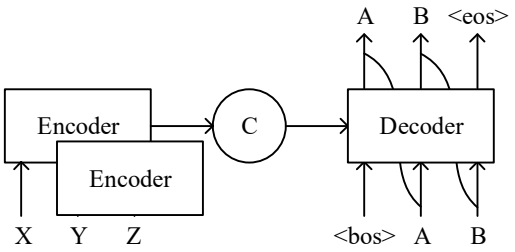


Figure 1: Encoder - decoder structure

II. B. 2) Text Feature Representation

Machine translation, as a kind of natural language processing task, first needs to convert the original language input into digital vectors that can be processed by computers, a process called Embedding word embedding.

In the natural language processing task, Embedding word embedding is text feature representation. Text feature representation is the most core and basic part of the natural language understanding process, and accurate text feature representation is the key to ensure that the machine translation task can be decoded smoothly and generate accurate translations.
vector. However, the simple encoding of unique hot vectors using only “0” and “1” will cause “semantic gap”, and its sparsity will bring about the sparseness of the encoding will bring “dimensionality disaster”. Therefore, based on the distributed assumption that words in similar contexts have similar meanings, some researchers have proposed word vector models based on distributed representations, such as CBOW, Skip-Gram, Word2vec, GloVe, LSTM.

Recurrent Neural Networks (RNN) are commonly used to process sequence data, and for text sentences, sentence features can be obtained after iterating the words in the sentence in a loop. However, because recurrent neural networks share a set of model parameters, the gradient will tend to 0 or infinity during chain derivation, leading to the problem of gradient vanishing and gradient explosion. The Long Short-Term Memory (LSTM) neural network has gradually become a representative of recurrent neural networks. LSTM decouples the originally shared model parameters through input gates, forgetting gates and output gates [22].

II. B. 3) Greedy Search vs. Cluster Search

The two most commonly used decoding methods for machine translation in the decoding stage are greedy search and cluster search.

Greedy search implies that the model selects the maximum probability for each word in word-by-word decoding, if the output sequence of the decoder is $\hat{Y}=(\hat{y}_1,...,\hat{y}_T)$, the process of the decoder greedily searching for the output $\hat{y}_t$ at the moment of t is shown in Eq. (17), where V denotes the target language word list.

$$\hat{y}_t=\arg\max_{y\in V}\log p(y \mid \hat{y}_{0\sim t-1}, x_{1\sim T};\theta) \tag{17}$$

The application of cluster search can alleviate the above problems to a certain extent and make the translation results closer to the global optimum. Unlike greedy search which selects the word with the largest generated probability at each moment, cluster search first determines a cluster width, and at each decoding, the results of the number of cluster widths are retained and selected in order of probability. Essentially, cluster search may be similar to greedy search, but cluster search expands the search space by caching the sequence of cluster width candidates and selecting the one with the largest integrated probability as the output, which results in a more diversified

translation result and makes the generated translation result close to the global optimum. Combined with equations (18), (19) and (20), the process of cluster search decoding is briefly explained here:

$$C_{t-1} = \{\tilde{y}_{0\sim t-1}^{(1)}, \tilde{y}_{0\sim t-1}^{(2)}, ..., \tilde{y}_{0\sim t-1}^{(K)}\} \tag{18}$$

$$C_t = \{\tilde{y}_{0\sim t}^{(1)}, \tilde{y}_{0\sim t}^{(2)}, ..., \tilde{y}_{0\sim t}^{(K)}\} = \arg sort^{\kappa} \sum_{t=0}^{t} \log p(y_t \mid y_{0\sim t'-1}, x_{1\sim T'}; \theta) \tag{19}$$

$$\hat{Y} = \arg\max_{\hat{Y}_1,...,\hat{Y}_K} (\frac{1}{|Y|})^{\alpha} \log p(Y \mid X; \theta) \tag{20}$$

(1) Let the cluster width be K, then the cluster candidate sequences at the $t-1$ moment are shown in Eq. (18), with a total of K sequences of length $t-1$.

(2) When t moments, K greedy searches are performed for K candidate sequences respectively, as shown in Eq. (19).

(3) Whenever there is a sequence output, this sequence decoding is finished, until all K sequences are decoded.

(4) For the generation of K candidate sequences, such as formula (20), the use of taking the logarithmic regularization method for reordering, in which $|Y|$ represents the length of the sequence, when α to take 1, formula (20) calculated that the perplexity of the value of the actual general take α for 0.6, can effectively avoid the generation of too short a sequence.

II. C.Neural Machine Translation Evaluation Metrics

BLEU index is actually to calculate the similarity between the translation result and the reference translation, when the machine translation result is closer to the reference translation, the corresponding BLEU value is higher. BLEU is calculated through an n-gram language model to calculate the proportion of the n meta-phrases of the machine translation that corresponds to the proportion of the reference translation occupied in the reference translation, and usually take n as 1~4, and take the average value of the result obtained by taking n equal to 1~4 as the comprehensive evaluation result. The specific calculation method of BLEU is shown in Eqs. (21) and (22):

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} w_n \log p_n) \tag{21}$$

$$BP = \begin{cases} 1, c > r \\ e^{(1-r/c)}, c \le r \end{cases} \tag{22}$$

where BP is the length penalty factor, N is the maximum order taken by n, w_n is the weight coefficient, and p_n corresponds to the proportion of the n meta-phrase occupied in the reference translation.

III. Neural Machine Translation Model for Engineering English Education

III. A. Domain Adaptive Techniques and Data Enhancement

III. A. 1) Domain adaptive techniques

Domain Adaptation is a technique in the field of machine learning that aims to solve the problem of models trained in one specific domain performing poorly on another domain. In machine learning, it is often assumed that training data and test data are obtained from the same distribution, but in real situations, this assumption is often not true. For example, in a language recognition task, the training data may come from domain data such as news and radio, but in the testing phase, the input data may be telephone recordings. This difference in data distribution may lead to a degradation of the performance of the trained model in the testing phase.

To solve this problem, domain adaptive techniques are introduced for related tasks. The aim of domain adaptive techniques is to make the model perform well on data from different domains by adjusting its feature representation. Domain adaptive mainly improves from two aspects, data and model. Where model improvement mainly operates from the aspects of model weights as well as model structure.

When domain-specific training data is scarce, the performance of neural machine translation tends to degrade. Migrating the NMT system to a specific domain can improve the performance of the model in a specific domain, but unfortunately, not all domains have a large amount of high-quality bilingual data. Domain-adaptive techniques data-wise improvements are made to improve domain translation by utilizing different data sources, such as forward and

backward translation techniques, selecting data based on embedded similarity, or improving the order of data in the training and fine-tuning phases [23].

III. A. 2) Data enhancement

In view of the fact that the boundaries of terms are more obvious, which is conducive to the analysis of the overall structure of the sentence, this paper is mainly based on the framework of "extraction and reorganization" to generate a pseudo-parallel corpus, and a complete example of the method is shown in Figure 2.

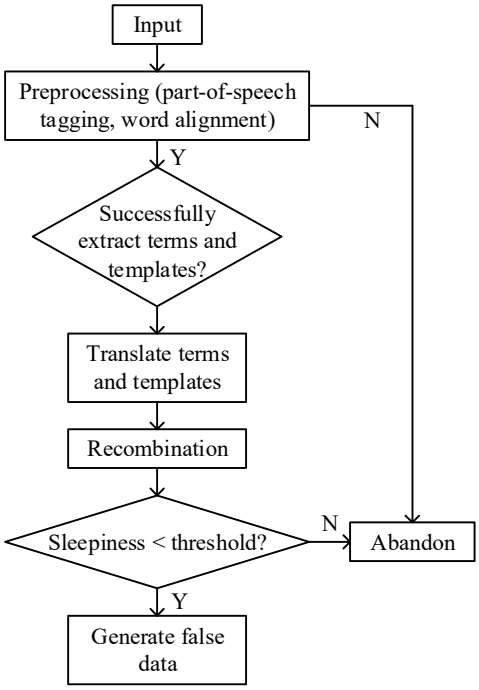


Figure 2: Data enhancement flowchart

The specific process of the data enhancement technique described in this paper is as follows:

(1) Perform lexical annotation and word alignment on the input text.

(2) Extract terms and templates from the labeled text according to the rules based on the Chinese text; if the extraction is successful, the English end of the text is generated based on the word alignment results of the corpus; if the extraction is not successful, it is discarded.

(3) Reorganize the terms and templates extracted from the corpus to generate a pseudo corpus.

(4) Calculate the perplexity of the pseudo-corpus and keep the corpus for those less than the threshold, otherwise discard it.

III. B. Neural Machine Translation Model for Fusing Underlying Information

The original Transformer translation model directly uses the output of the top layer of the encoder as the feature vector of the source language for decoding, which will lead to a certain extent to the lack of the underlying information of the source language contained in the feature vector, therefore, this paper improves the structure of the Transformer model on the basis of the Transformer model, adds an additional feature extraction layer before the model encoder, and respectively uses a variety of We add an extra feature extraction layer before the model encoder, and use various network structures to realize the feature extraction of the underlying information of the source language, and finally transmit the extracted information to the output of the top layer encoder unit, and use the residual connection to fuse the two vectors, and then send the fused feature vector as the final representation vector of the source language to the decoder side for the subsequent decoding and translation process. And the decoder side is improved by using the residual connection structure to transmit the target language word embedding vectors at the bottom of the decoder side to the output of the top layer decoder unit, so as to realize the transmission of the bottom layer information at the decoder side, and to play a greater advantage of the bottom layer information on the basis of fusion of the bottom layer features at the encoder side, so as to make full use of the bottom layer features of the target language in the decoding process.

III. C. Feature Extraction Layer

III. C. 1) Feedforward networks

In order to realize the function of feature extraction by neural network, this paper firstly uses two-layer feedforward neural network Feedforward for feature extraction of source language information with the following formula:

$$c_{feed\_out} = w_{fn2}(\mathrm{Re}\,lu(w_{fn1}c_{emb})) \tag{23}$$

where $w_{fn1} \in R^{d*d}, w_{fn2} \in R^{d*d}$, d are the word vector dimensions set within the Transformer model.

III. C. 2) BiLSTM networks

Long and short-term memory network belongs to a variant of recurrent neural network, which is mainly aimed at the problems of gradient vanishing and gradient explosion existing in recurrent neural network, and updates and controls the information transfer process through its internally designed state vectors and each gating mechanism.

The LSTM internally contains three gating mechanisms, the forgetting gate f_t, the input gate i_t, and the output gate o_t, and an internal state vector c_t. where:

(1) The forgetting gate f_t is used to control how much information is forgotten about the internal state vector c_{t-1} of the previous moment;

(2) Input gate i_t is used to control how much information is saved for the candidate state $\tilde{c}_t$ of the current moment;

(3) Output gate o_t for controlling how much information the internal state vector c_t of the current moment outputs to the output h_t of the current moment.

The state vector c_t consists of the state vector c_{t-1} of the previous moment and the candidate state $\tilde{c}_t$ of the current moment, and then nonlinearly passes its internal information to the hidden layer vector h_t of the output of the current moment through the output gate o_t, which is calculated as follows:

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{24}$$

$$h_t = o_t \odot \tanh(c_t) \tag{25}$$

The calculations for each gating mechanism and candidate unit $\tilde{c}_t$ in the above equation are as follows:

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c) \tag{26}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{27}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{28}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{29}$$

where x_t denotes the input at the current moment, h_{t-1} denotes the hidden state at the previous moment, σ is the Sigmoid activation function, and both W, U and b are parameters to be learned.

In this paper, the BiLSTM network used to construct the feature extraction layer is stacked using a two-layer LSTM network structure, and the source language input to the network is feature extracted from both the left-to-right and right-to-left directions at the same time, and then the source language feature vectors obtained from both the forward and backward directions are spliced together in the last dimension to obtain the output vector of the BiLSTM network.

In this paper, the hidden layer dimension inside the BiLSTM network is set to be consistent with the translation model, but after the BiLSTM network will lead to its output word vector dimension is not consistent with the translation model, so in order to make the two dimensions consistent, this paper adds a feed-forward neural network at the output end of the BiLSTM network for dimensional transformation in the word vector dimension, as shown in the following equation.

$$c_{bilstm\_out} = w_{fn\_bilstm} * c_{bilstm} \tag{30}$$

where $w_{fn\_bilstm} \in R^{2d \times d}$, d refer to the word vector dimensions set within the Transformer translation model.

III. C. 3) BiGRU network

The gated recurrent unit network can be regarded as a simplified version of the LSTM network, whose internal structure no longer uses memory units, but only uses two gate mechanisms for information screening and transmission, and can still achieve similar effects as the LSTM network.

The GRU network internally contains two gating mechanisms, update gate z_t and reset gate r_t, both computed from the input x_t of the current moment and the hidden state h_{t-1} of the previous moment, and nonlinearly computed using the Sigmoid activation function to control the value between 0 and 1, with the following formula:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z) \tag{31}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r) \tag{32}$$

The candidate state $\tilde{h}_t$ inside the network is obtained from the input x_t of the current time step and the hidden state h_{t-1} of the previous moment, and the hidden state of the current time step of the output consists of the candidate state $\tilde{h}_t$ inside the network and the hidden state h_{t-1} of the previous moment, as in the following equation:

$$\tilde{h}_t = \tanh(W_{xh} x_t + r_t \odot (U_{hh} h_{t-1}) + b_h) \tag{33}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t \tag{34}$$

where σ is the Sigmoid activation function, and W, U and b are both parameters to be learned.

In this paper, both BiGRU network and BiLSTM are set up to use stacked structure to extract features from the source language input in both positive and negative directions at the same time, and set the dimensions of its internal hidden vectors to be consistent with the word vector dimensions set up by the Transformer, and the same feed-forward neural network is used to transform the output vectors of the network in terms of the word vector dimensions as shown in Eq. (35).

$$c_{bigru\_out} = w_{fn\_gru} * c_{bigru} \tag{35}$$

where $w_{fn\_bigru} \in R^{2d \times d}$, d refer to the word vector dimensions set within the Transformer translation model.

III. C. 4) Multi-Head Self-Attention Network

In addition to the work on the underlying feature extraction of utterances using the above network architecture, this paper also attempts to use the same multi-head self-attention mechanism as the Transformer model to form the feature extraction layer for the underlying feature extraction of the source language. The self-attention mechanism is able to extract the features inside the sentence well by distributing the attention weights among the words inside the sentence, and the weights of the attention mechanism are computed using the deflated dot product.

The self-attention mechanism with multiple heads, on the other hand, means that the query vector Q and the key-value pairs K and V are first transformed in the word vector dimension by means of linear mapping when performing the self-attention mechanism computation, and the original representation vectors are divided into query vectors and key-value pairs of small dimensions in their own word vector dimension, respectively, and then the attention mechanism is performed on the obtained h sets of vectors, and finally the obtained h heads of attention weights are spliced and mapped to reduce them back to the original word vector dimension to get the final attention allocation.

In order to prevent too much stacking of the layers of the multi-head self-attention mechanism, which leads to the problem of overfitting and information loss in the model, this paper refers to the internal structure of the Transformer and uses the Transformer's internal encoder unit as the extraction layer of the underlying features, and adopts the same masking mechanism as that of the Transformer's encoder end within this feature extraction layer.

IV. Experimental design and comparative analysis

IV. A. Experimental design

IV. A. 1) Experimental data

The exact number of experimental datasets in this chapter is shown in Table 1, where the training set, test set as well as validation set are divided as shown in Table 2.

Table 1: Pseudo Parallel Data Tables Generated Using Different Data Augmentation Methods

Initial corpus	Simple data enhancement	Translation	Two-way iteration	Alternate training iteration
35w	65w	65w	65w	65w

Table 2: Specific partitioning of the dataset

Corpus type	Training set	Verification set	Test set
D	35w	6000	6000
D1	93w	11000	11000
D2	93w	11000	11000
D3	93w	11000	11000
D4	93w	11000	11000

IV. A. 2) Experimental configuration and training

The details of the experimental environment are shown in Table 3, and the specific training process consists of three parts.

The first part is to use the BERT pre-training model as the semantic encoder, which is a powerful multilingual pre-training language model for natural language processing tasks on multiple languages. In terms of parameter settings, the standard configuration of BERT-Base is used, where the number of encoder layers is set to 12, the number of attention heads is 12, the number of hidden units is set to 768, and the learning rate is set to 0.005.

In the second part, Transformer was used as a neural machine translation model. The number of encoder and decoder layers of this model is 6, the number of attention heads is 8, the dropout ratio is set to 0.1, and the initial learning rate is 0.001. In addition, the ReLU activation function is chosen to enhance the nonlinear characteristics of the model during the training process. During the training process of the translation model, the cross-entropy loss function is used to optimize the model to enhance the accuracy of translation.

The third part is to train a neural machine translation model with semantic context data augmentation model, which first samples semantic vectors in the optimized semantic space and fuses the semantic vectors with the original sentence vectors. This translation model is optimized during the training process by optimizing the loss in the contrast learning process with the cross-entropy loss in the Transformer model.

Table 3: Experimental Environment and Related Configuration

Experimental environment	Relevant configuration
Operating system	Ubuntu 16.04.6 LTS
GU	Nvidia Tesla P100
Python	3.7.10
CUDA	10.2
PyTorch	1.8.1

IV. B. Performance Comparison of Neural Machine Translation for Engineering English Education

IV. B. 1) Parameter adjustment

At the encoder side of the Transformer model, the more layers of the encoder, the more semantic information can be learned. In this paper, in order to verify the effect of the number of encoder layers on the translation performance of the model, experiments are conducted on the Chinese-English dataset and the English-French dataset, and the results of the experiments are shown in Fig. 3, Fig. 4. When the number of decoder layers is less than 6, the translation performance of the translation model rises with the increase of the number of layers, and when the number of decoder layers is greater than or equal to 6, the BLEU value tends to stabilize gradually. So according to the experimental results, the number of decoder layers in this paper is set to 6.

IV. B. 2) Losses during training and changes in accuracy

In order to more intuitively observe the trend of the loss and accuracy change of the translation model proposed in this paper during the training process, a visual display diagram is provided as shown in Figure 5. By observing Fig. 5, (a) and (b) show the loss as well as accuracy changes of the model during the training process, respectively, the dynamic changes of the model during the training process as well as the optimization process can be deeply understood.

From the figure, it can be seen that in the early stage of model training, the loss function decreases quite rapidly, which indicates that the model is able to learn the distribution and patterns of the data quickly in the initial stage. At

the same time, the accuracy of the translation also improves rapidly at this stage, indicating that the model starts to effectively translate the source language into the target language. This rapid progress is due to the initialization of the model parameters and the choice of optimization algorithms, which enable the model to achieve better performance in the early stages of training.

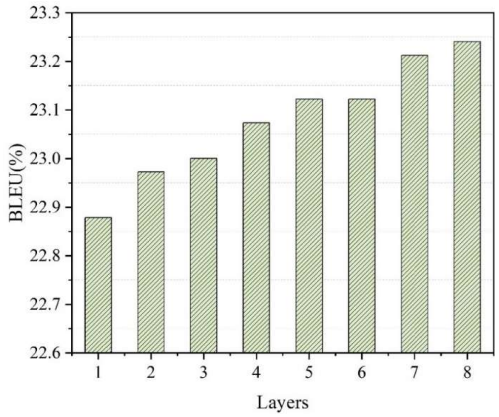


Figure 3: Chinese-English translation results

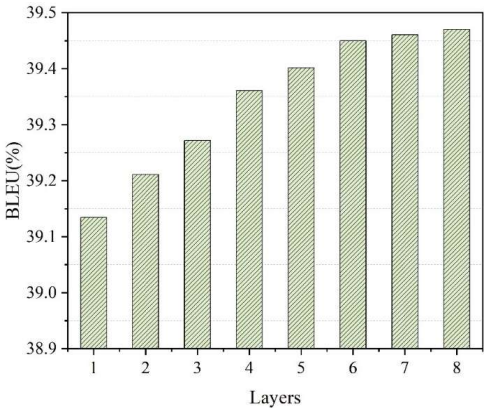
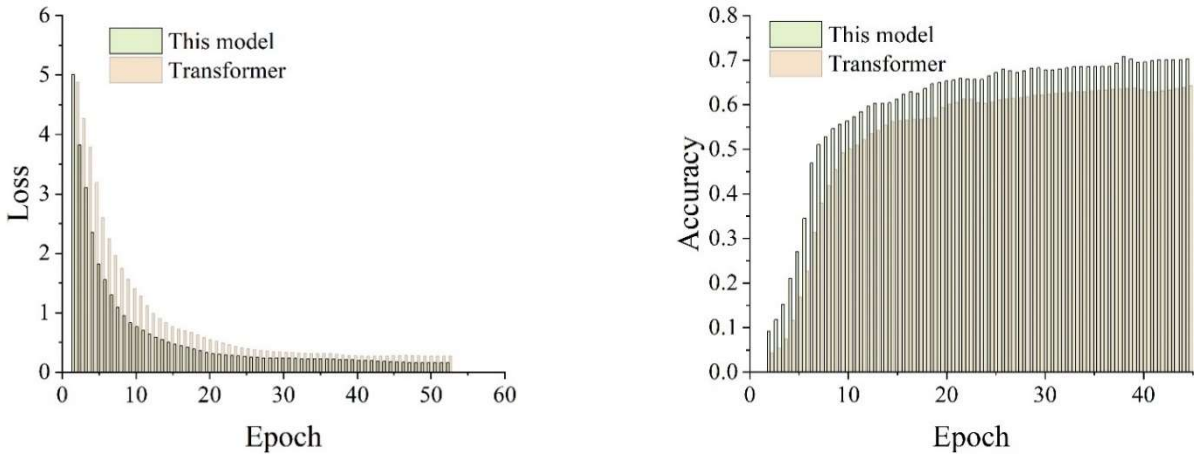


Figure 4: English-French translation results



(a) Loss (b) Accuracy

Figure 5: Loss and Accuracy Variation During Training Process

As the training time increases, the change in loss, while still continuing, has decayed relatively little and has gradually leveled off. This indicates that the model has gradually reached a stable state in the later stages of training, and is no longer experiencing large performance gains as it did in the early stages. However, this smooth state does not mean that the model has stopped learning, but indicates that the model is able to reduce the risk of overfitting while maintaining a certain level of accuracy. At the same time, the accuracy of the model's translations continues to increase during the training process. When a certain value is reached, the accuracy rate starts to stabilize, which means that the model has reached the upper limit of its performance. This steady increase in accuracy rate not only demonstrates the stability of the model, but also proves the effectiveness of the translation model in this paper. By comparing the base Transformer model, the advantage of this paper's translation model in convergence speed as well as accuracy rate can be clearly seen. This enhancement is not only reflected in the numerical value, but also in the generalization ability and robustness of the model. By introducing the data enhancement method, the translation model in this paper effectively alleviates the semantic inconsistency problem in the process of data enhancement and improves the translation quality of the model.

IV. B. 3) Machine Translation Attention Heat Map

In order to see more clearly the attentional weights of the source and target languages in each part of the translation process, a heat map is used here to illustrate the final decoding effect, as shown in Figure 6. For easy observation, the engineering English text is converted to Latin alphabet here, and its correspondence is [illegible](tede)[illegible](ene)[illegible](turshiltai) [illegible] (yi)[illegible] (kikhu)[illegible] (ni)[illegible] (jokhistai)[illegible] (geghu) [illegible] (ta). The darker the color of the squares in this figure means the higher the probability of the candidate words it provides, and the highlighted area in the diagonal line indicates that the model correctly aligns the words or phrases of the input and output sequences during the translation process, which indicates that the model focuses on the Chinese words in the corresponding position in most cases when translating each engineering English word, which also proves the accuracy of the model's translation.

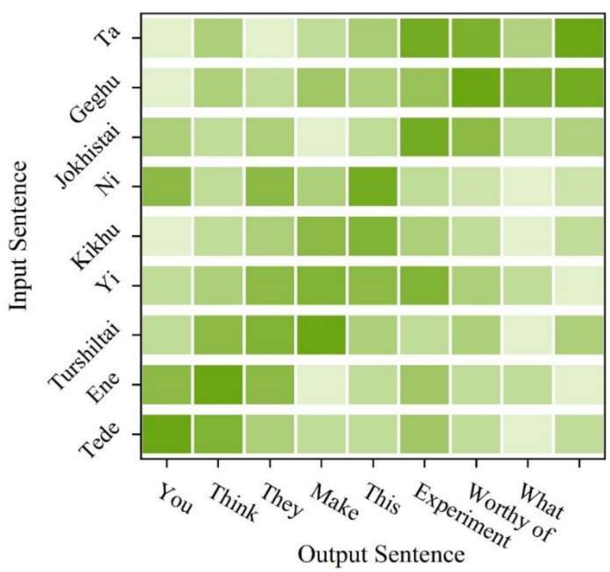


Figure 6: Heat map in machine translation

IV. B. 4) Comparison of translation performance for different word frequencies and sentence lengths

(1) Comparison of Translation Performance of Sentences of Different Lengths

In order to verify the effect of sentence length on the translation performance of the proposed model in this paper, this paper randomly selects 5200 pairs of data in the Chinese-English dataset. And these data are divided by sentence length and compared with the translation model Transformer. In order to visualize the trend of the direction of translation performance, this paper draws a histogram of the translation results as shown in Figure 7.

From the figure, it can be seen that when the sentence length increases from 15 to 25, the translation model can learn more semantic information, and the translation performance is improved, when the sentence length exceeds 25, if the sentence length continues to increase, the performance of the translation model shows a decreasing trend, for the neural machine translation model that incorporates the information field of engineering English, the underlying information introduced can capture long-distance information, and in the case of the sentence length

between 45 to 55, the translation effect of the model is significantly better than the traditional translation model, indicating that the model in this paper is better than the Transformer model in capturing long-distance information.

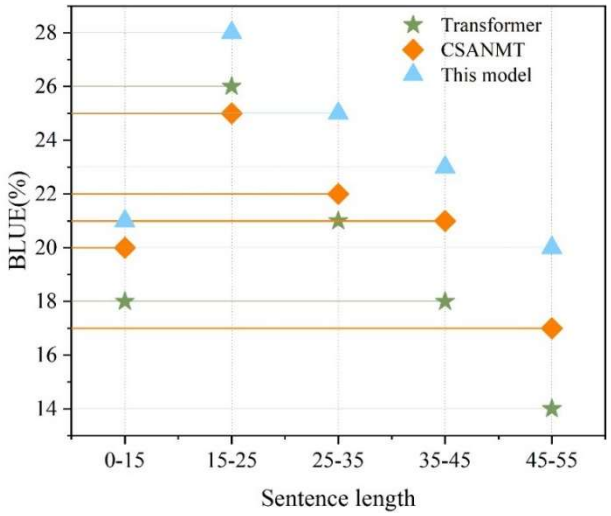


Figure 7: Comparison of translation performance of sentences of different lengths

(2) Translation accuracy under different word frequencies

Figure 8 shows the translation accuracy corresponding to the frequency of different word occurrences under the Transformer model and the translation model approach of this paper. As expected results, the generalization performance of the translation model proposed in this paper for low-frequency words is better than the base Transformer model. This suggests that the translation model proposed in this paper can effectively alleviate the problem of poor translation quality for low-frequency words by successive enhancement of the semantics during the training process.

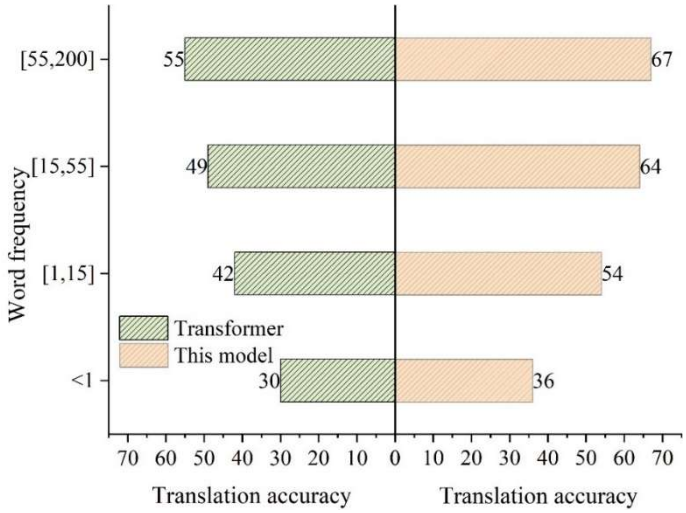


Figure 8: Translation accuracy of words under different word frequencies

In order to explore the translation effect of the method in other low-resource languages, the adopted dataset is tested on the data enhancement methods in this chapter as well as other data enhancement methods. The specific translation results are shown in Table 4. It can be clearly seen through the table that the translation modeling approach proposed in this paper not only achieves satisfactory translation results in neural machine translation tasks, but also shows the same strong performance and potential when facing low-resource translation tasks like Chinese-French and English-French.

Table 4: The translation results of this model in different data sets

Method	China-English	China-French	English-French
Transformer	32.18	21.9	32.77
SwitchOut	32.24	22.26	32.98
MixSeq	32.27	22.23	33.06
SCA	32.94	22.79	33.44
CSANMT	32.7	23.18	33.86
CMLM	33.15	22.95	33.71
This article	34.65	23.7	34.39

IV. C. Analysis of experimental results

IV. C. 1) Comparative experiments

In order to further validate the effectiveness of the method proposed in this paper, comparative experiments are conducted with other models on the electrical engineering corpus, and the experimental results are shown in Table 5. The experiments in this paper use the optimal parameter settings, i.e., applying the feedforward neural network only in the 2nd, 4th, and 6th layers of the encoder, and setting the number of sublayers of the decoder to 2.

From the experiments of the Transformer-Big and Re-Transforme models, it can be seen that although increasing the width of the model can improve the model performance and the quality of the translation to a certain extent, it also increases the training time significantly, and the Transformer-Big takes the longest time, which is 96.2 minutes longer than Model 1, with a 89.37% increase in the time spent, but the BLEU does not improve as much as the model in this paper. It can also be found that these improvement methods, which are more classical in the general domain, have limited performance improvement for translation tasks in the electrical domain, and the BLEU of Re-Transformer not only does not keep the same as Model 1, but even decreases. It can be seen that the translation quality of these comparison models within the electrical engineering domain is not as good as the model proposed in this paper.

Table 5: Contrast test

Model	BLEU/%	Blediff value/%	Time/min	Time gain/%
Model 1	35.43	-	112.4	-
Model 2	35.71	0.71	144.3	32.53
Model 3	35.58	0.58	147.5	34.41
Model 4	36.17	0.87	121.4	1.11
Model 5	35.01	0.32	137.3	24.65
Model 6	35.26	0.49	112.8	2.51
Transformer-Big	36.49	1.12	208.6	89.37
Re-Transformer	34.37	-0.26	81.4	-26.73
This model	36.79	1.18	119.5	7.29

IV. C. 2) Ablation experiments

In order to verify the value of the hyperparameter θ, as well as the effect of the way the matrix replicas are combined on the performance of the model proposed in this paper, ablation experiments were carried out on the above contents, and the results are shown in Fig. 9 and Table 6. The content shaped as $(A,B)\times C$ in the table indicates that both H_D^A and H_D^B in H_D are replicated as C copies.

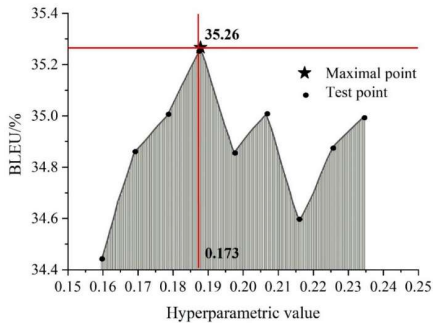


Figure 9: Experiment of ultrafilre ablation

Table 6: Replication strategy ablation experiment

Numbering	Combination strategy	BLEU/%	Blediff value/%
1	$(1,6)\times3\|(3,4)\times2\|(2,5)\times1$	36.29	-
2	$(3,4)\times4\|(2,5)\times2\|(1,6)\times1$	35.55	-0.52
3	$(2,5)\times3\|(3,4)\times2\|(1,6)\times1$	35.29	-0.89
4	$(3,4)\times3\|(1,6)\times2\|(2,5)\times1$	35.43	-0.62
5	$(2,5)\times3\|(1,6)\times2\|(3,4)\times1$	36.17	-0.24
6	$(1,6)\times3\|(2,5)\times2\|(3,4)\times1$	35.09	-0.83

Figure 9 shows the results of the ablation experiment for θ. In principle, the translation model in this paper uses 515-dimensional feature dimensions, and assuming that all the feature dimensions of a word in T_P contribute exactly the same amount to terminological information enhancement, the value of each feature dimension of that word in the calculation of $Soft\max$ should be: $\frac{1}{515}=0.001942\approx0.002$.

Therefore, this paper conducts experiments with 0.002 as the benchmark, and the results show that the position greater than 0.0019 in T_P is the most favorable for terminological information reinforcement, when θ becomes larger, it means the screening mechanism is more strict, and vice versa, it means the screening mechanism is more lenient, and both cases will lead to a decrease in the quality of the translation.

Table 6 shows the experimental results under different replication strategies. In the multilayer attention model, the first and the last layers usually extract the positional and proper noun features of the language, while the middle layer has the most consistent dependency features. As a discrete vocabulary, there is no more important information than the distributional features of lexical meaning as well as location under a specific task, so in this paper, we combine layers 1 and 6 in 6 layers H_D; layers 3 and 4; and layers 2 and 5, and change the data volume share between the combinations through replication to help the model focus on more valuable feature information. The results show that the best replication strategy is $(1,6)\times3\|(3,4)\times2\|(2,5)\times1$, with data volume shares of 52%, 36% and 19%, respectively.

The term vocabulary in the term dictionary is generated by BI-LSTM-CRF, in order to verify whether the generated term words are accurate and sufficient, we designed the ablation experiment, and the experimental results are shown in Figure 10. The terminology dictionary used in the model of this paper contains 7520 terminology vocabularies, and this paper takes this as the benchmark, and reduces the number of vocabularies to 7000, 6500, 5500 and 5000 respectively when deleting; and adds the number of vocabularies to 8000, 8500, 9000 and 9500 respectively when adding, because the terminology vocabularies extracted from BI-LSTM-CRF only have 7520 terminology vocabularies. Since there are only 7520 terms extracted by BI-LSTM-CRF, the added terms are from non-terminal words in the English lexicon. The process of term deletion and addition is a randomized process, and a random seed is set in the experiment to ensure that the results can be reproduced, and two sets of experiments are carried out in this paper to ensure that the results are statistically reliable by changing the random seed.

Analyzing the graphs, it can be seen that any addition and deletion of words in the terminology dictionary will damage the BLEU of the model in this paper. adding words to the terminology dictionary will cause the model to include many non-terminal words into the terminology training, and the model will be difficult to learn and summarize the features and laws of the terminology, which is not conducive to accurate translation of real terminology words, and the more words are added additionally, the more the terminology words will be diluted, and the lower the BLEU of the model will be. BLEU will be lower. Deleting words in the terminology dictionary will cause the model to ignore part of the terminology words for training, and cannot extract these terminology features, not to mention the reinforcement of the terminology information, as the deletion process is stochastic, when the high-frequency terminology words are deleted, it will have a large impact on the BLEU, and vice versa, the impact on the BLEU is small, therefore, there is no obvious law between the number of deletion words and the BLEU, but the overall view is always detrimental to the model's BLEU is always harmful. In summary, it can be proved that the vocabulary in this paper's terminology dictionary is appropriate and suitable.

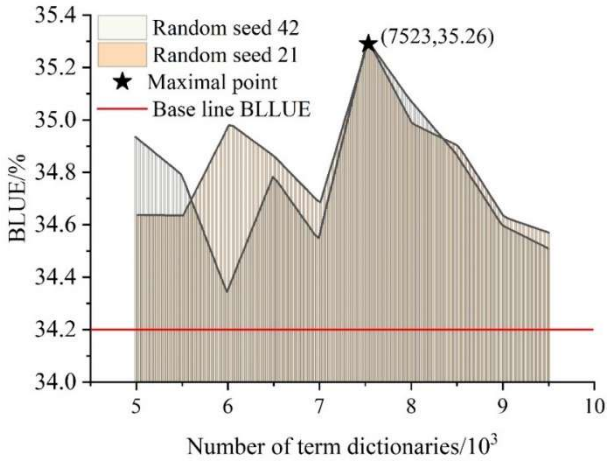


Figure 10: The term quotient ablation experiment

V. Conclusion

In this paper, based on the translation model, an extra underlying feature extraction layer is added at the bottom of the encoder side, and four network structures, Feedforward, BiLSTM, BiGRU and Multi-Head Self-Attention, are used to extract the underlying information of the source language. After that, the obtained underlying features containing more syntactic information are fused with the feature vectors containing more semantic information output from the top-level encoder unit to improve the adaptability to the translation model in engineering English education. It is proved experimentally that this method can significantly improve the performance of engineering text translation tasks, and the BLEU value of the translation model is improved by 1.40 compared with the Transformer baseline model. Then the model is applied to two low-resource datasets, Chinese-French and English-French, and the experimental results also show significant improvement. Meanwhile, compared with other translation models, the model in this paper takes into account the translation quality and training time, realizes the comprehensive improvement of the performance of the translation model in the engineering domain under the low-resource condition, and provides a new solution idea for balancing the contradiction between terminology information enhancement and low-resource corpus training.

Funding

Key Research Project of Higher Education Teaching Reform in Jiangsu Province (2023JSJG227);
Education and Teaching Project of China Construction Education Association (2023326);
Research Project on Quality Assurance and Evaluation of Higher Education in Jiangsu Province (2023 - Y27);
Special Project of "Double High - level Plan" of Jiangsu Vocational Institute of Architectural Technology (SGZX2022 - 1).

References

[1] Van den Beemt, A., MacLeod, M., Van der Veen, J., Van de Ven, A., Van Baalen, S., Klaassen, R., & Boon, M. (2020). Interdisciplinary engineering education: A review of vision, teaching, and support. Journal of engineering education, 109(3), 508-555.
[2] Broo, D. G., Kaynak, O., & Sait, S. M. (2022). Rethinking engineering education at the age of industry 5.0. Journal of Industrial Information Integration, 25, 100311.
[3] Leydens, J. A., & Lucena, J. C. (2017). Engineering justice: Transforming engineering education and practice. John Wiley & Sons.
[4] Kamp, A. (2023). Engineering education in the rapidly changing world: Rethinking the vision for higher engineering education. TU Delft, WL/Delft Hydraulics.
[5] Zhu, M. (2022). Factors influencing analysis for level of engineering english education based on artificial intelligence technology. Mathematical Problems in Engineering, 2022(1), 4447209.
[6] Srivani, V., Hariharasudan, A., Nawaz, N., & Ratajczak, S. (2022). Impact of Education 4.0 among engineering students for learning English language. PLoS One, 17(2), e0261717.
[7] Rus, D. (2020). Creative methodologies in teaching English for engineering students. Procedia Manufacturing, 46, 337-343.
[8] Faxriddinovna, K. B. (2023). Theoretical views of teaching English to engineering students. The American Journal of Social Science and Education Innovations, 5(12), 113-116.
[9] Yasmin, M., & Yasmeen, A. (2021). Viability of outcome-based education in teaching English as second language to chemical engineering learners. Education for Chemical Engineers, 36, 100-106.
[10] Alemi, M. (2016). The role of technical english language on modern engineering education. Majallah-i Amuzih-i Muhandisi-i Iran, 18(69), 1.
[11] Saienko, N., Olizko, Y., & Arshad, M. (2019). Development of tasks with art elements for teaching engineers in English for specific purposes classroom. International Journal of Emerging Technologies in Learning (iJET), 14(23), 4-16.

[12] Kuswoyo, H., Sujatna, E. T. S., Rido, A., & Indrayani, L. M. (2020, September). Theme choice and thematic progression of discussion section in engineering English lectures. In Proceedings of the 4th International Conference on Learning Innovation and Quality Education (pp. 1-10).
[13] Bishop, C. M., & Bishop, H. (2023). Deep learning: Foundations and concepts. Springer Nature.
[14] Wang, X., Zhao, Y., & Pourpanah, F. (2020). Recent advances in deep learning. International Journal of Machine Learning and Cybernetics, 11, 747-750.
[15] Kamath, U., Liu, J., & Whitaker, J. (2019). Deep learning for NLP and speech recognition (Vol. 84, p. 1). Cham, Switzerland: Springer.
[16] Otter, D. W., Medina, J. R., & Kalita, J. K. (2020). A survey of the usages of deep learning for natural language processing. IEEE transactions on neural networks and learning systems, 32(2), 604-624.
[17] Zhang, G. (2020). A study of grammar analysis in English teaching with deep learning algorithm. International Journal of Emerging Technologies in Learning (iJET), 15(18), 20-30.
[18] Tao, X. (2021, April). Ways to promote students' deep learning in English teaching based on computer technology. In Journal of Physics: Conference Series (Vol. 1881, No. 2, p. 022042). IOP Publishing.
[19] Lang, A. (2022). Evaluation algorithm of English audiovisual teaching effect based on deep learning. Mathematical Problems in Engineering, 2022(1), 7687008.
[20] Zhai, Q. (2023). A Study on the Design of English Language Teaching Activities Based on Deep Learning. Advances in Vocational and Technical Education, 5(3), 46-52.
[21] Khenglawt Vanlalmuansangi,Laskar Sahinur Rahman,Pakray Partha & Khan Ajoy Kumar. (2024). Addressing data scarcity issue for English–Mizo neural machine translation using data augmentation and language model. Journal of Intelligent & Fuzzy Systems(3),6313-6323.
[22] Filipa S. Barros,Paula A. Graça,J.J.G. Lima,Rui F. Pinto,André Restivo & Murillo Villa. (2024). Using Recurrent Neural Networks to improve initial conditions for a solar wind forecasting model. Engineering Applications of Artificial Intelligence(PC),108266-.
[23] Marcus Tomalin,Bill Byrne,Shauna Concannon,Danielle Saunders & Stefanie Ullmann. (2021). The practical ethics of bias reduction in machine translation: why domain adaptation is better than data debiasing. Ethics and Information Technology(3),1-15.