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

Combinatorial Press

# Research on Syntactic Optimization and Semantic Reconstruction Strategies for English Translation Based on Machine Learning

Yongjia Huang[1,] ✉
[1] *School of Foreign Languages, Zhengzhou Shengda University, Zhengzhou, Henan, 451191, China*

ABSTRACT
The development of globalization has contributed to the increasing demand for cross-language communication, and machine translation, as an effective language conversion tool, has improved the quality and efficiency of English translation. The article discusses the syntactic optimization and semantic reconstruction strategies for English translation based on machine learning. The machine translation model of English syntax optimization and semantic reconstruction based on EM algorithm is constructed by using key technologies such as EM algorithm and multi-head attention mechanism. The model adopts a joint learning method, combining the Transformer model with the EM algorithm. The dependency between any two words in the input sequence is captured using the multi-head attention mechanism, and the new translation corpus is generated by multi-task joint training algorithm. The training phase of this paper's model has good translation effect, and the model of this paper gets the highest BLEU score of 32.86 when the number of multi-head attention layers is 1. The distribution of semantic features of translation reconstruction under this paper's method is basically consistent with the simulation results, and the error elimination rate of semantic reconstruction is 99.64% when the number of samples is 500. The method in this paper is more effective in syntactic structure optimization, with the highest BLEU scores on “Chinese to English” and “English to Chinese”, and the syntactic correctness rate on English long sentences of different topics reaches 88.69%~96.57%.

*Keywords:* machine learning, EM algorithm, multiple attention, Transformer model, English translation

## 1. Introduction

In today's era of globalization, the importance of English is self-evident, and it has become a global common language. As an important communication tool, translation plays an important role in promoting communication between different languages and cultures [1-4]. English translation is the

✉ Corresponding author.
*E-mail address:* huang0105chen@163.com (Y. Huang).
Received 12 January 2024; Revised 05 March 2024; Accepted 25 December 2024; Published Online 15 April 2025.
DOI: 10.61091/jcmcc127a-266

process of transforming information from English language to another language. English translation is not only simply translating words and sentences, but also needs to consider the meaning of words, grammatical structure, context, and cultural background and other factors [5-8], and machine learning can play an important role in the process of English translation in terms of syntax and semantics of translation [9].

In recent years, with the popularization of the Internet and the acceleration of globalization, language translation technology has become a research that has attracted much attention in the field of science and technology. In this context, machine learning technology is widely used in the field of language translation, which becomes an important means to realize the automation of language translation [10-13]. Machine learning-based language translation technology refers to the use of machine learning models to automate the translation of texts between different languages. This kind of translation no longer needs human translation, but by training the machine learning model, the source language text is transformed into the target language text [14-17]. At present, the language translation techniques based on machine learning are mainly categorized into two kinds: statistical machine translation and neural machine translation. Among them, statistical machine translation is an earlier proposed translation technique, which mainly utilizes probabilistic models for translation, while neural machine translation is a new technique developed in recent years, which uses deep learning features for translation [18-21].

The study exploits the powerful generalization performance of both EM algorithm and Transformer model in English machine translation to generate a high-quality English translation corpus by jointly learning the model. Meanwhile, the Transformer model adopts a multi-head attention mechanism to understand and represent the input text from multiple perspectives. Maximum likelihood-based principle is used to complete the determination of the model training objective function, aiming to optimize the English translation syntax and accurately achieve semantic reconstruction. Meanwhile, in order to verify the effectiveness of the method, the study utilizes the IWSLT and MTNT4 datasets to conduct translation model training, syntax optimization, and semantic reconstruction experiments.

## 2. Key technologies

### *2.1. EM algorithm under machine learning*

The EM algorithm [22] can be described as the following two steps: the E (Estimate) step and the M (Maximization) step, and a loop iteration is performed based on these two steps.The EM algorithm is essentially a data addition algorithm. Each iteration of the algorithm consists of two steps: the first step is to find the conditional expectation of the log-likelihood function (E-step), and the second step is to maximize the conditional expectation obtained from the E-step computation (M-step). It uses data expansion to reduce the more complex likelihood function optimization problem into a series of optimization problems for simpler functions.

In general, denote by $p(\theta \mid Y)$ the likelihood density function of the parameter $\theta$ to be estimated, which is called the distribution density function of the posterior distribution, $p(\theta \mid Y, Z)$ the density function of the posterior distribution about $\theta$ obtained after adding the data $Z$, and $p(Z \mid \theta, Y)$ the density function of the conditional distribution of the latent data $Z$ given $\theta$ and the observed data $Y$ and use this conditional density function to adjust for the added data, the purpose of the previous work is to compute the value of $p(\theta \mid Y)$ at the estimation of the unknown parameter $\theta$ at extreme values, so next the whole algorithm is run as follows:

Noting $\theta^{(i)}$ as the estimate of the a posteriori plurality at the beginning of the $i+1$ nd iteration, the two steps of the $i+1$ rd iteration are the E-steps. Place $p(\theta \mid Y, Z)$ or $\log p(\theta \mid Y, Z)$ on the hidden data $Z$ to compute its conditional expectation so that the hypothesized hidden data can be eliminated, i.e:

$$Q(\theta \mid \theta^{(i)}, Y) = E_z[\log p(\theta \mid Y, Z) \mid \theta^{(i)}, Y] = \int \log[p(\theta \mid Y, Z)] p(Z \mid \theta^{(i)}, Y) dZ \tag{1}$$

Step M: Maximize $Q(\theta \mid \theta^{(i)}, Y)$ and find a point $\theta^{(i+1)}$ to make:

$$Q(\theta^{(i+1)} \mid \theta^{(i)}, Y) = \max_{\theta} Q(\theta \mid \theta^{(i)}, Y) \tag{2}$$

An iterative process $\theta^{i} \to \theta^{i+1}$ is thus formed, iterating the above E-steps and M-steps until they are sufficiently small:

$$\| \theta^{i} \to \theta^{i+1} \| \text{ or } \| Q(\theta^{(i+1)} \mid \theta^{(i)}, Y) - Q(\theta^{(i)} \mid \theta^{(i)}, Y) \| \tag{3}$$

When the observed data follow a Gaussian normal distribution, the likelihood density function of the fully observed data can be expressed as:

$$P(\theta \mid Y, Z) = (\sqrt{2\pi}\sigma)^{-n} e^{\frac{(l_1-\mu)^2}{2\sigma^2}} \cdot e^{\frac{(l_2-\mu)^2}{2\sigma^2}} \cdots\cdots e^{\frac{(l_n-\mu)^2}{2\sigma^2}} \tag{4}$$

Where $\theta$ is the parameter to be determined $(\mu, \sigma)$, $Y$ are the known observations $(l_1, l_2 \cdots\cdots, l_{n-1})$, $Z$ are the missing data $l_n$. Missing data $l_n$ The probability density function of the conditional distribution based on the known observations and parameters is:

$$P(Z \mid \theta^{i}, Y) = (\sqrt{2\pi}\sigma_i)^{-1} e^{\frac{(l_n-\mu_i)^2}{-2\sigma_i^2}} \tag{5}$$

where $\mu, \sigma_i$ is the parameter estimate after the $i$ nd iteration.The expectation step of the EM algorithm is calculated as follows:

$$Q(\theta \mid \theta^{i}, Y) = Ez[\ln P(\theta \mid Y, Z) \mid \theta^{i}, Y] = \int \ln P(\theta \mid Y, Z) \cdot P(Z \mid \theta^{i}, Y) dZ$$
$$= \int \left( -\frac{n}{2}\ln(2\pi) - n\ln\sigma + \frac{(\sum_{i=1}^{n}(l_i-\mu)^2)}{-2\sigma^2} \right) o\left(\sqrt{2\pi}\sigma_i\right)^{-1} e^{\frac{(l_n-\mu_i)^2}{-2\sigma_i^2}} dl_n \tag{6}$$

The resulting $Q(\theta \mid \theta^{i}, Y)$ after integration is maximized to give $\theta^{i+1}$. Since $Q(\theta \mid \theta^{i}, Y)$ integrates out the missing data, the method of finding the extremes is the same as for the calculation of the maximum likelihood estimate. Substitute $\theta^{i+1}$ to recalculate the expected value, and so on until $\| \theta^{i} + 1 - \theta^{i} \|$ or $\| Q(\theta^{i+1} \mid \theta^{i}, Y) - Q(\theta^{i} \mid \theta^{i}, Y) \|$ is sufficiently small. At the end of the iteration, the best valuation of the unknown parameter with missing data is obtained.

### *2.2. Attention mechanisms*

2.2.1. Encoder-Decoder Framework. For understanding the attention mechanism, one has to first understand the encoder-decoder framework [23]. An encoder-decoder is a Seq2Seq model of sequences, i.e., a model that outputs a Y at the decoder side by inputting a sequence X at the encoder side. The encoding process in the encoder-decoder model is to transform the input sequence into a vector with a fixed length. The decoding process, on the contrary, is to decode the fixed-length vector output from the encoding side into an output sequence, which theoretically represents exactly the same meaning or data.

The field of machine translation is generally an LSTM-LSTM encoder-decoder framework and so on, so it is a framework-type model rather than a concrete model. Its input and output parts can be any form of data, including text, images, video, formula data, etc. One of its more abstract frame representations is shown in Figure 1 below.

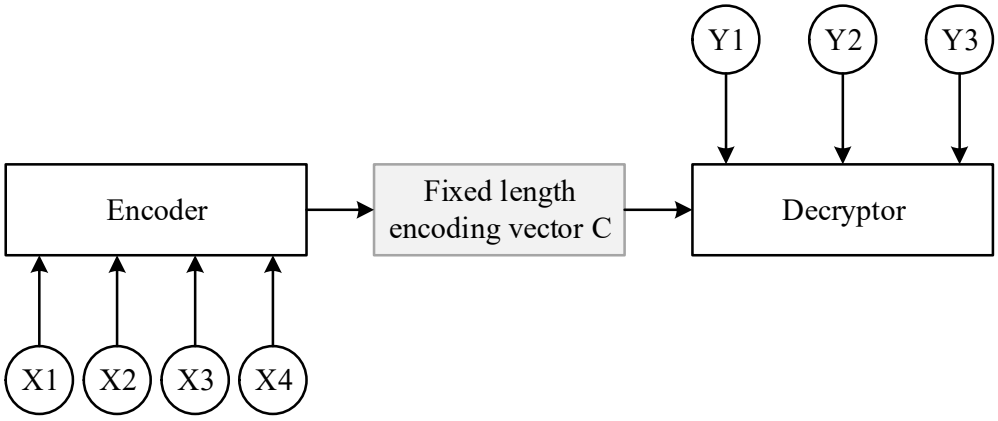


**Fig. 1.** Abstract encoder-decoder framework

2.2.2. Attention module. Figure 2 shows the encoder-decoder framework with the addition of the attention model. In order to solve the problems that exist above, it is necessary to introduce the attention model [24], and the key to understanding the attention model is that the fixed intermediate semantic representation C is replaced with C1 that is adapted to the changes added to the attention model based on the current output word.

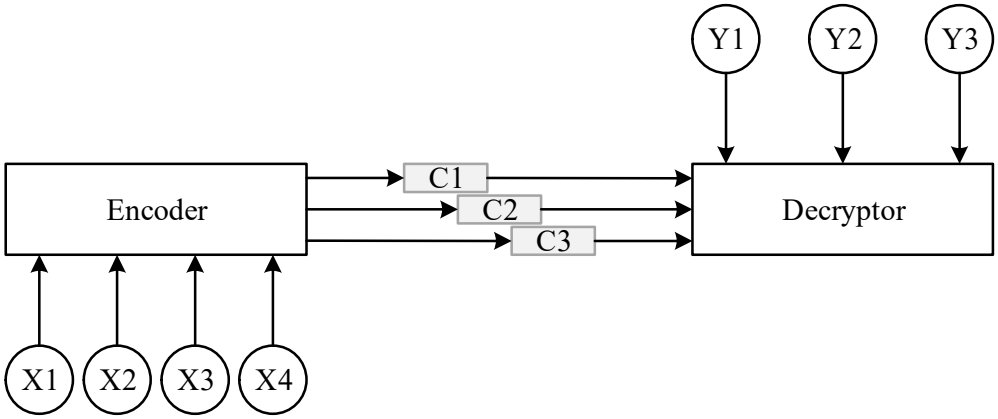


**Fig. 2.** Encoder-decoder framework with attention model

At this point the word $y_i = g(C_i, y_1, y_2, \cdots, y_{i-1})$ of moment $i$ is generated, where:

$$C_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j \tag{7}$$

$T_x$ denotes the length of the input sequence, $\alpha_{ij}$ denotes the attention allocation weight of the $j$ th word in the input sentence at the time of outputting the $i$ rd word, and $h_j$ is the semantic encoding of the $j$th word in the input sentence.

2.2.3. Self-attention mechanisms. The self-attention mechanism [25] involves three vectors: a query vector, a key vector and a value vector. An attention matrix and attention weights are used for each word in the input sequence, so that each word in the sequence has a bidirectional representation, that is, it contains information about the word above and information about the word below. The attention mechanism first inputs the semantic vector representation of the target word and each word in the context, and obtains the query vector representation of the target word, the key vector representation of each word in the context, and the original value vector representation of the target word and each word in the context by linear transformation, and computes the query vectors and each key vector in

order to obtain the similarity weights, and the commonly used similarity functions are dot product, splicing, and perceptron, among others. These weights are then normalized using the softmax function. Finally, the weights are summed with the original value vectors of the target words to obtain the value vectors of each word in the context, i.e., the final semantic vector representation. As the output of Attention. The computational procedure is shown below:

$$Q, K, V = liner(Q, K, V) \tag{8}$$

$$f(Q, K_i)\begin{cases} Q^T K_i & dot \\ Q^T W_\alpha K_i & geneal \\ W_\alpha [Q; K_i] & concat \end{cases} \tag{9}$$

$$W_i = soft\max(f(Q, K_i)) = \frac{\exp[f(Q, K_i)]}{\sum_{j=1}^{J} \exp[f(Q, K_i)]} \tag{10}$$

$$Self - Attention(Q, K, V) = \sum_{j=1}^{J} W_i V_i \tag{11}$$

Where, $Q$ is the word in the sentence, $K$ is the individual word in the context, $V$ is the respective original value in both the target word as well as the context word, $K_i$ is the key value of the $i$ th word, $W_i$ is the weight vector of the $i$ th word, $f(Q, K_i)$ is the similarity, Self-Attention is the probability distribution of the attention, $j$ denotes the number of dimensions, and $J$ denotes an upper bound on the number of dimensions.

In order to enhance the diversity of Self-Attention, multi-head Self-Attention is further utilized, that is, different Self-Attention modules obtain the augmented semantic vectors of each word in the text under different semantic spaces, and linearly combine multiple augmented semantic vectors of each word to obtain a final augmented semantic vector.

## 3. Joint Learning Strategy for English Syntax Optimization and Reconstruction Based on EM Algorithm

### *3.1. Co-training of Transformer based on EM algorithm*

3.1.1. Transformer model. The structure of the Transformer model is given in Fig. 3, where the encoding part is composed of 6 layers of encoders, and similarly, the decoding part is composed of 6 layers of decoders.

Each layer of the encoder consists of a self-attentive computation sublayer and a feed-forward neural network (FNN) sublayer, each of which is configured with a residual connection to implement the regularization operation. The decoder consists of three sublayers per layer. The first sublayer still uses the Self-Attention computation layer, which differs from the Self-Attention layer in the encoder in that the decoder adds a mask to the Self-Attention vectors in order to block the influence of the following information in the computation. The role of the second sublayer is to compute the attention vector between the output of the encoder and the state of the decoder's hidden layer. The third sublayer is the FNN sublayer, which serves the same purpose as the encoder.

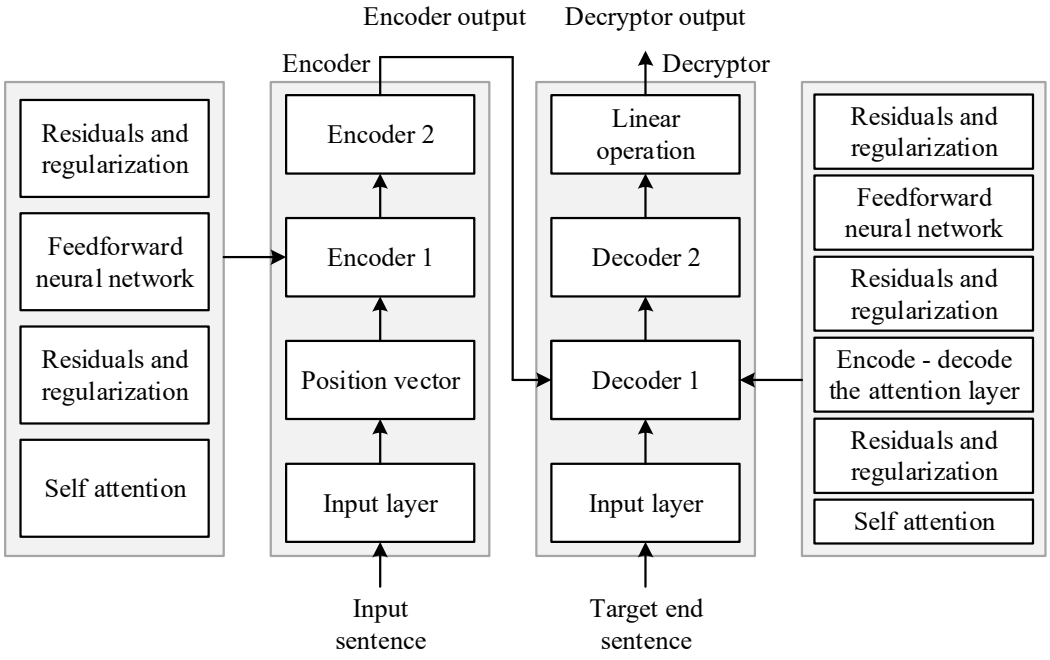


**Fig. 3.** Transformer model structure

The Transformer model works as described below [26]:

1) Compute the input sentence vector representation.

2) For each word in the sentence, the word vectors $e_i$, where $e_i \in R^{l\times p}$, are multiplied by three weight matrices of dimension $p \times d$, denoted $W^Q, W^K, W^V$, respectively, to obtain the corresponding three $d$-dimensional vectors, denoted $q, k, v$.

3) Transformer uses the multi-head attention mechanism to matrix splice $k$ and $v$ of each word to get the large matrix $K, V, n$ of the spliced $n \times d$ as the number of words, and the number of heads used here is $m$ for the convenience of generalization.

4) According to the multi-head attention mechanism, slice $q, K, V$ to get $\{q_i\}_{i=1}^m, \{K_i\}_{i=1}^m, \{V_i\}_{i=1}^m$ respectively.

5) For any $i \in m$, compute:

$$C_i = attention(q_i, K_i, V) = soft\max(\frac{q_i K_i^T}{\sqrt{d}})V \tag{12}$$

6) Perform multiple attention calculations:

$$MultiHead(q, K, V) = Concat(\{C_i\}_{i=1}^m)W^O \tag{13}$$

Take the calculation of layer $l$ of the encoder side of the Transformer model as an example, where $l > 1$. The working principle of the model is explained:

(1) In the self-attention mechanism module, according to the calculation results above:

$$h_i^{l,c} = MultiHead(h_i^{l-1}, \{h_j^{l-1}, h_j^{l-1}\}_{j=1}^n) \tag{14}$$

(2) In the residual linking and normalization part, layer normalization is utilized in the original text and is calculated as follows:

$$\eta_i^{l,c} = LayerNorm(h_i^{l,\varepsilon} + h_i^{l-1}) \tag{15}$$

(3) After the layer normalization calculation, the nonlinear transformation using the next fully connected layer is calculated as follows:

$$h_i^{l,n} = \mathrm{ReLU}(\eta_i^{l,\varepsilon} W_1^x + b_1^x) W_2^x + b_2^x \tag{16}$$

(4) Finally, the results above are then computed using the pressed residual join and layer normalization modules to obtain the hidden state representation $h_i^l$, which is computed as follows:

$$h_i^{'} = LayerNorm(h_i^{l,n} + \eta_i^{l,\varepsilon}) \tag{17}$$

In the above computation, layer normalization facilitates training. Specifically, let $z$ denote a $d$-dimensional vector, i.e., $z = [z_1, z_2, ..., z_d]$. The layer normalization is computed as follows:

$$LayerNorm(z) = \left[ \frac{z_1 - \mu}{\sqrt{\sigma^2 + \varepsilon}}, \frac{z_2 - \mu}{\sqrt{\sigma^2 + \varepsilon}}, ..., \frac{z_d - \mu}{\sqrt{\sigma^2 + \varepsilon}} \right] \tag{18}$$

$$\mu = \frac{1}{d} \sum_{i=1}^{d} z_i \tag{19}$$

$$\sigma^2 = \frac{1}{d} \sum_{i=1}^{d} (z_i - \mu)^2 \tag{20}$$

where $\varepsilon$ denotes the hyperparameter to prevent the case where the denominator is zero. The following is an explanation of the computation process for the hidden state $s_i^l$ at moment $i$ of the decoder:

(1) The bottom part of the decoder is the attention mechanism module of the mask, which is calculated as follows:

$$s_i^{l,s} = MultiHead(s_i^{l-1}, \{s_j^{l-1}, s_j^{l-1}\}_{j=1}^{i}) \tag{21}$$

(2) Computation of the first residual connection and layer normalization module:

$$\Delta_i^{l,\varepsilon} = LayerNorm(s_i^{l,\varepsilon} + s_i^{l-1}) \tag{22}$$

(3) The following is the computation of the source-to-target-language attention mechanism module, which takes into account the information at the source-language end, but the computation also employs a multi-head attention mechanism in the following process:

$$s_i^{l,c} = MultiHead(\Delta_i^{l,s}, \{h_j^L, h_j^L\}_{j=1}^{n}) \tag{23}$$

(4) Computation of the second residual join and layer normalization module:

$$\Delta_i^{l,c} = LayerNorm(s_i^{l,c} + \Delta_i^{l,\varepsilon}) \tag{24}$$

(5) Computation of fully connected networks:

$$s_i^{l,n} = \mathrm{ReLU}(\Delta_i^{l,c} W_1^t + b_1^t) W_2^t + b_2^t \tag{25}$$

(6) Finally, there is a third residual connection and the computation of the layer normalization module, and finally, the hidden state $s_i^l$ at moment $i$ of the decoder is computed as:

$$s_i^l = LayerNorm(s_i^{l,n} + \Delta_i^{l,c}) \tag{26}$$

3.1.2. Joint learning models. In this paper, we utilize the joint learning method [27], taking the generation of Chinese-English parallel corpus as the main task, the training of Transformer model as the auxiliary task, combining the main and auxiliary tasks through EM algorithms, and adopting the joint training method to generate large-scale and high-quality parallel corpus by using the existing corpus. In the process of corpus generation, the Transformer model is combined with the EM

algorithm, and the parameters are tuned through iterative training until the whole process converges. The joint learning model architecture is shown in Figure 4.

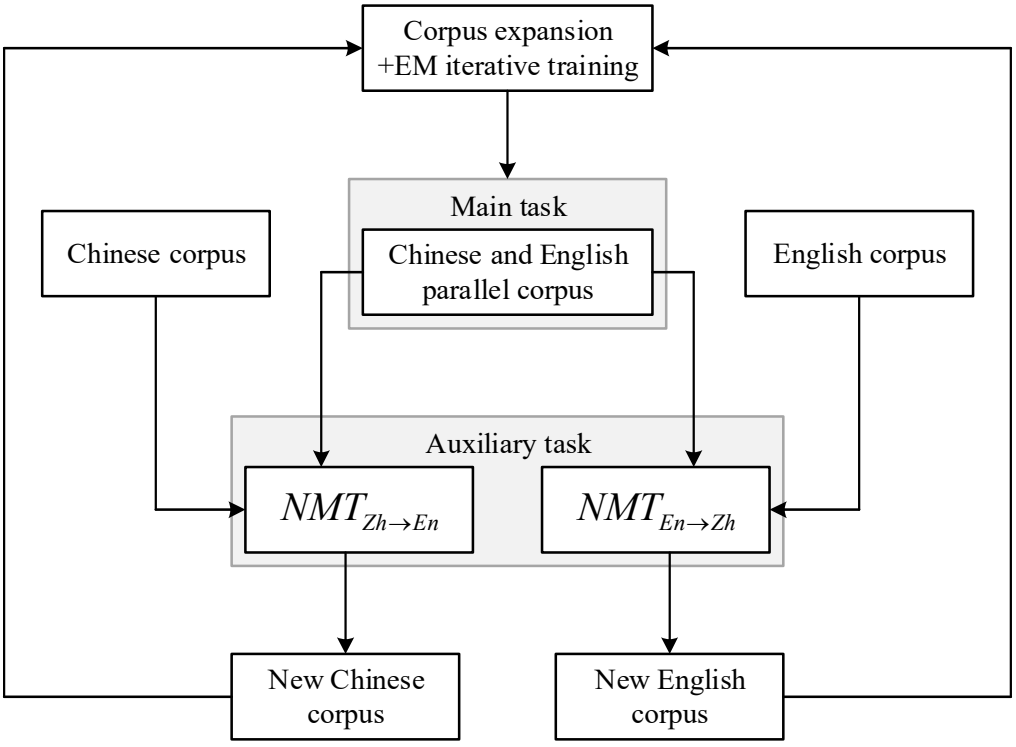


**Fig. 4.** Joint learning model architecture

3.1.3. Joint multi-task training algorithm:

1) Pre-training of word vectors. In this paper, we utilize publicly available parallel corpus resources and online Chinese-English monolingual corpus resources as the initial corpus resources, and carry out pre-processing work on the corpus, including pre-processing of the source and target languages. The Word2Vec model is used to generate word vectors, which are used as inputs to two unidirectional Transformer models to initialize the training model.

2) EM iterative training algorithm based on the Transformer model. In this paper, the Chinese-to-English Transformer model is labeled as $NMT_{Zh\to En}$, and the English-to-Chinese model is labeled as $NMT_{i:n\to z_n}$. The goal of iterative training of the models is to jointly optimize $NMT_{Zh\to En}$ and $NMT_{En\to Zh}$ using the Chinese corpus $S$ and the English corpus $T$. The specific algorithms for iterative training are shown below:

(1) Initialize $NMT_{Zh\to En}$ and $NMT_{En\to Zh}$ using random weight parameters $\theta_{S\to T}$ and $\theta_{T\to S}$.

(2) Pre-train $NMT_{Zh\to En}$ and $NMT_{En\to Zh}$ on the bilingual parallel corpus $D=\{(S_i,T_i)\}_{i=1}^{N}$. Send generations using the EM algorithm until convergence as described below:

E-Step:

For $NMT_{Zh\to En}$, use $NMT_{En\to Zh}$ to generate translation $\hat{S}'$ using the bulk English corpus $T'=\{t_i\}_{i=1}^{p}$ and combine the two to form a new parallel corpus $D'=\{T',\hat{S}'\}$. Similarly, for $NMT_{En\to Zh}$, use $NMT_{Zh\to En}$ to generate translation $\hat{T}'$ using the bulk Chinese corpus $S'=\{S_j\}_{j=1}^{q}$ and combine the two to form a new parallel corpus $D'=\{S',\hat{T}'\}$.

M-Step:

For $NMT_{Zh\to En}$, the weighted bilingual corpus $DUD'$ is utilized to train $NMT_{Zh\to En}$, where the weights of the generated corpus are assigned to be from the translation probability $P(s\mid t_j)$, and the weights of the real corpus are assigned to be 1. Similarly, for $NMT_{En\to Zh}$, the weighted bilingual corpus $DUD'$ is utilized to train $NMT_{En\to Zh}$, where the weights of the generated corpus are assigned to be

from the translation probability $P(t \mid s_j)$, and the weights of the real corpus are assigned to be 1.

### *3.2. Objective function for model training*

For the pre-training process, this is done using an approach based on the maximum likelihood principle, which has the general approach of maximizing the log conditional probability of a correct translation, given the model parameters of the source language sentence $\theta$, with the goal of obtaining a $\theta$ that satisfies the following equation:

$$\theta^* = \arg\max_\theta \sum_{i=1}^{N} \sum_{j=1}^{|t_i|} \log p(t_{i,j} \mid t_{i,0\to j-1}, s_i) \tag{27}$$

where: $N$ is the size of the training corpus and $|t_i|$ is the length of the target language sentence $t_i$.

For a given original parallel corpus $D=\{(S_i,T_i)\}_{i=1}^{n}$ and English monolingual corpus $T'=\{t_i\}_{i=1}^{p}$, the training objective of the model is to maximize the likelihood of bilingual data and monolingual data, i.e:

$$L^*(\theta_{S\to T}) = \sum_{i=1}^{N} \log p(t_i \mid s_i) + \sum_{i=1}^{p} \log p(t_i) \tag{28}$$

Where: the first part represents the conditional probability of generating English $T$ for Chinese $S$. The second part represents the language model about $T$. The decomposition of the language model part for English is obtained:

$$\log\, p(t_i) = \log \sum_s q(s) \frac{p(s,t_i)}{q(s)} \tag{29}$$

According to Jensen's inequality, the following equation can be obtained:

$$\log \sum_s q(s) \frac{p(s,t_i)}{q(s)} \geq \sum_s q(s) \log \frac{p(s,t_i)}{q(s)} \tag{30}$$

$$\sum_s q(s) \log \frac{p(s,t_i)}{q(s)} = \sum_s \left[ q(s) \log\, p(t_i \mid s) - KL(q(s) \square\, p(s)) \right] \tag{31}$$

where: $s$ represents the hidden variable of the Chinese translation generated by the English sentence $t_i$, $q(s)$ is the approximate probability distribution of $s$, $p(s)$ represents the marginal distribution of $s$, and $KL(q(s) \| p(s))$ is the $KL$ divergence between the two probability distributions. The following equation can be applied when $q(s)$ meets the following conditions:

$$\frac{p(s,t_i)}{q(s)} = c \tag{32}$$

where: $c$ is constant and does not depend on $t$, for given:

$$\sum_s q(s) = 1 \tag{33}$$

Then $q(s)$ can be calculated by the following equation:

$$q(s) = \frac{p(s,t_i)}{c} = \frac{p(s,t_i)}{\sum_s p(s,t_i)} = p^*(s \mid t_i) \tag{34}$$

Where: $p^*(s \mid t_i)$ denotes the optimal translation probability from English to Chinese. Using the translation probability $p(s \mid t_i)$ 2 as the approximate probability distribution of the source language satisfies the conditions of the following equation:

$$p(s \mid t_i) \approx q(s) \tag{35}$$

Therefore, it can be obtained:

$$L^{*}(\theta_{S\to T}) \geq L(\theta_{S\to T}) \tag{36}$$

$$L(\theta_{S\to T}) = \sum_{i=1}^{N} \log p(t_i \mid s_i) + \sum_{j=1}^{T} \sum_{s} [p(s \mid t_j) \log p(t_j \mid s) - KL(p(s \mid t) \square p(s))] \tag{37}$$

The above equation shows that $L(\theta_{S\to T})$ serves as a lower bound for the optimal translation $L^{*}(\theta_{S\to T})$ and that $KL(p(s \mid t_j) \| p(s))$ is independent of the parameters $\theta_{s\to\tau}$ of the model. So the optimization function $L(\theta_{S\to\tau})$ can be further simplified:

$$L(\theta_{S\to T}) = \sum_{i=1}^{N} \log p(t_i \mid s_i) + \sum_{j=1}^{T} \sum_{s} p(s \mid t_j) \log p(t_j \mid s) \tag{38}$$

Similarly, the optimization objective function for English to Chinese translation model $NMT_{En\to Zh}$ can be obtained:

$$L(\theta_{T\to S}) = \sum_{i=1}^{N} \log p(s_i \mid t_i) + \sum_{j=1}^{S} \sum_{t} p(t \mid s_j) \log p(s_j \mid t) \tag{39}$$

where: $t$ is used as a hidden variable of the source language sentence $S_j$. The training objective function of the joint learning model is obtained as:

$$L(\theta) = L(\theta_{S\to T}) + L(\theta_{T\to S}) \tag{40}$$

## 4. Experimental configuration and analysis of results

### *4.1. Experimental configuration*

4.1.1. Experimental data. In order to verify the effectiveness of the method, the experiments in this paper mainly train the translation model on parallel datasets by the International Workshop on Spoken Language Translation (IWSLT for short), including IWSLT171, IWSLT152, and IWSLT133. In addition, in order to test the adaptive performance of the translation system on noisy text, the dataset MTNT4 (Testbed for Machine Translation of Noisy Text), which is specially used for the evaluation, is chosen for this paper. These noisy texts contain a variety of linguistic errors, including spelling mistakes, character omissions, character repetitions, character redundancies, grammatical errors, etc., and then they are translated by professional language users to obtain their correct translations.

4.1.2. Experimental setup. All experiments at the institute are based on fairseq, an open source framework from facebook. Transformer is used as the base model framework, the number of layers of both encoder and decoder is set to 5, the dimensions of word embedding and the dimensions of the output state of each layer are 512, the dimensions of the feedforward neural network module are set to 512, 2048, 512 in that order, and the number of heads of the multi-head attention module is set to 6. The model applies Dropout during training and label smoothing, the corresponding neuron deactivation rate and smoothing factor are set to 0.2, and the models for all language pairs are trained using a joint word list with a size of 20K. The learning rate change strategy during model training is inverse sqrt, the corresponding warm updates are set to 4000, and Adam is used as the optimizer. All the experimental data have been processed by word splitting, special symbol processing, and effective length sentence screening, with the ratio of parallel sentences set to 2 and effective lengths from 1-200.

4.1.3. Experimental environment. The experimental process of syntactic optimization and semantic reconstruction of English translation based on machine learning is done using Linux system and

Python language on a computer with NVIDIA GTX 1080ti GPUs with 11G of video memory, and the experiments are implemented on pytorch framework.

### *4.2. Experimental results and analysis*

4.2.1. Model evolution in the training phase. Syntactic optimization of English translation as well as semantic reconstruction model training evolution have been carried out for this paper's method, and the deep learning-based RNN-NMT method is chosen as a comparison.

In terms of evaluation metrics, BLEU values (Bilingual Evaluation Understudy) are used to judge the accuracy of Chinese-to-English neural machine translation, which replaces manual evaluation to measure the quality of machine translation.BLEU values indicate the degree of correspondence between the candidate (the sentence generated by the model to be evaluated) and the reference (the human translation as the reference answer). In other words, the BLEU value of a sentence indicates how well the candidate and the reference match. The BLEU value of a sentence is calculated as follows:

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} w_n \log P_n) \tag{41}$$

where $n$ denotes $n-gram$, i.e., the set of phrases of $n$ word lengths.

Another evaluation metric is the word alignment error rate (AER), which is calculated as follows:

$$\begin{aligned} Accuracy &= \frac{|A \cap P|}{|A|} \\ Recall &= \frac{|A \cap D|}{|D|} \\ AER &= 1 - \frac{|A \cap P| + |A \cap D|}{|A| + |D|} \end{aligned} \tag{42}$$

In the above formula, A denotes the set of output English words and Chinese words, D denotes the set of English words manually labeled as “sure” and Chinese words, and P denotes the set of English words manually labeled as “possible” and Chinese words. D is obviously a subset of P. In the specific experiments, this paper only adopts a “sure” labeling type, so D=P in this paper.

Figure 5 shows the changes of BLEU score and word alignment error rate of word granularity during the training process, the BLEU score is tested on the IWSLT corpus, and the word alignment error rate is tested on the MTNT4 corpus. As expected, the BLEU scores of both models gradually increase with training, and the Transformer-based machine translation model in this paper gets better translation results with BLEU scores between 22 and 28. However, the word alignment error rate of the Transformer machine translation model is higher than that of the recurrent neural network-based model, and it will become higher and higher as the training progresses, which means that the effect of word alignment will become worse and worse.

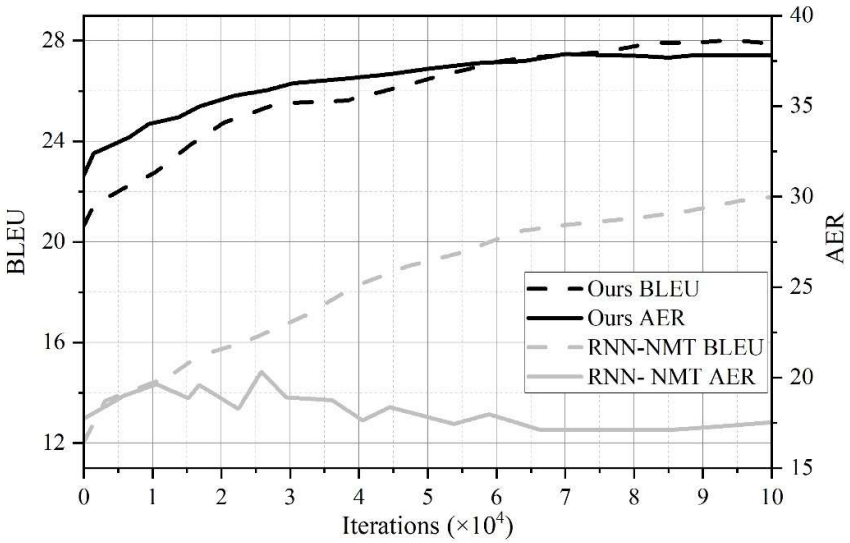


**Fig. 5.** The variation of the BLEU and the AER of the word alignment

This paper attempts to integrate knowledge of the syntactic structure of English at different layers of the multicentered attention mechanism in order to investigate the gap between the integration effects of different attention layers. In this section of the experiment, $\lambda_1$, $\lambda_2$ and $\lambda_3$ are all 0.60, 0.30 and 0.10 respectively. This experiment measured the BLEU values under different layers of multi-head attention mechanism five times and took the average value to ensure the accuracy of the experiment.

The BLEU scores under different layers of multi-head attention mechanisms are shown in Figure 6, where the horizontal coordinates indicate the BLEU values obtained using this paper's method when selecting the 1~8 layers of multi-head attention mechanisms for fusing English syntactic information.

The experimental results show that a relatively high BLEU value (32.86) was obtained by performing the fusion of the dependent syntactic information at the first layer of the multi-head attention mechanism. Compared with the lowest BLEU value obtained by performing fusion at the second layer, it improved by 5.45 points. This is because the first layer of the multi-head attention mechanism will pay more attention to the word to be translated itself, and introducing the contextual information of the parent word at the first layer can enable the word embedding to contain more features, thus obtaining a more effective word representation. This can further help the translation model to complete the translation process more accurately. Therefore, the appropriate attention layer is also conducive to improving the translation performance more effectively with the help of dependent syntactic information. The following experiments will be carried out when the number of layers of the multi-head attention mechanism is 1.

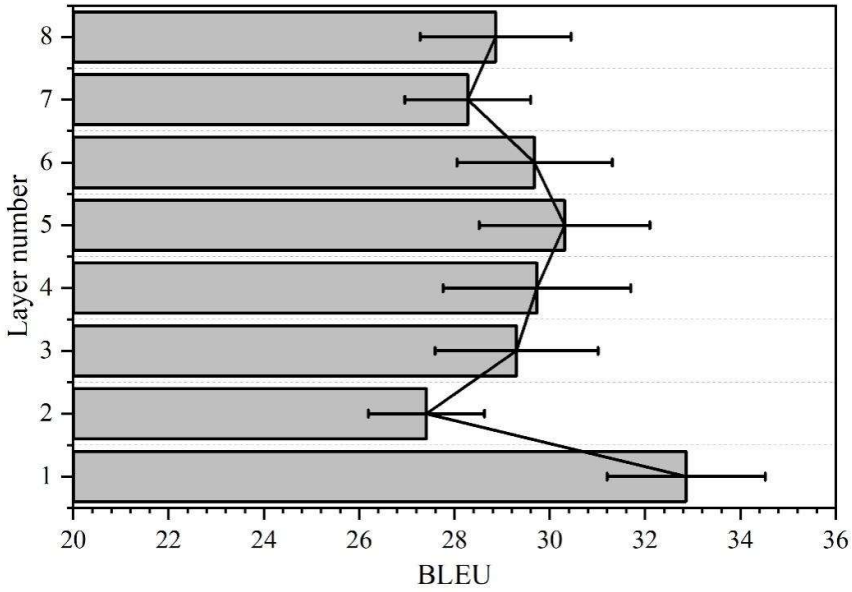


**Fig. 6.** The bleu scores under the same number of long attention mechanisms

4.2.2. Distribution of semantic reconstruction features. Experiments were conducted to verify the effectiveness of this paper's method in eliminating translation errors in English semantic reconstruction. The experiments are designed using Matlab, and the set of association rules for English conversion is represented in OAEI language. The set of simple semantic units is 1000, the set of sample training volume is 100, the number of iterations is 100, the set of semantic attributes is 50, the number of instances of similarity semantic feature distribution is 100, the number of learning times is 100, and the convergence step is 20.

According to the above simulation environment and parameter settings, the translation error in English conversion is eliminated and the distribution of semantic features is tested. Figure 7 demonstrates the feature distribution of the semantic feature simulation and semantic reconstruction calibration. As can be seen from the figure, the distribution of semantic features in the simulation is between [-1,1], and after the semantic reconstruction conversion by this paper's method, the distribution of the translated semantic features is still between [-1,1] and basically overlaps with the simulation results. It shows that the accuracy and correction ability of the method is good for eliminating errors in English conversion translation. The translation error elimination accuracy of English language conversion is high and the correlation of translation calibration is strong.

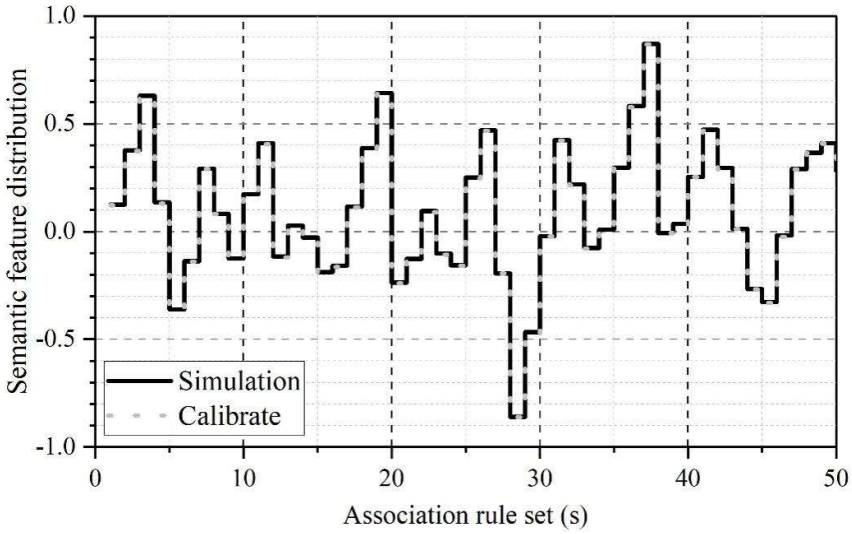

**Fig. 7.** The characteristics of simulation and semantic reconstruction calibration

In order to further evaluate the error elimination rate of this paper's method in semantic reconstruction, three commonly used semantic transformation methods are selected in this paper. These include multi-fuzzy semantic automatic judgment, particle swarm optimization algorithm, and deep learning error elimination method, which are denoted as Methods 1-3 in turn, and are compared with the results of this paper's method.

The comparison results of the semantic error elimination rate of each method are shown in Figure 8. It is clear from the figure that the error elimination rate of each method is improving with the increase of the number of test samples. When the number of samples is 500, the error elimination rate of multi-fuzzy semantic automatic judgment, particle swarm optimization algorithm, and deep learning error elimination method all reach the best, which are 91.09%, 90.35%, and 93.51%, respectively. And the method in this paper reaches 99.64%, which is 6.56% to 10.28% higher than the comparison methods. It indicates that the error elimination rate of semantic reconstruction of the proposed method in this study is higher than the commonly used methods.

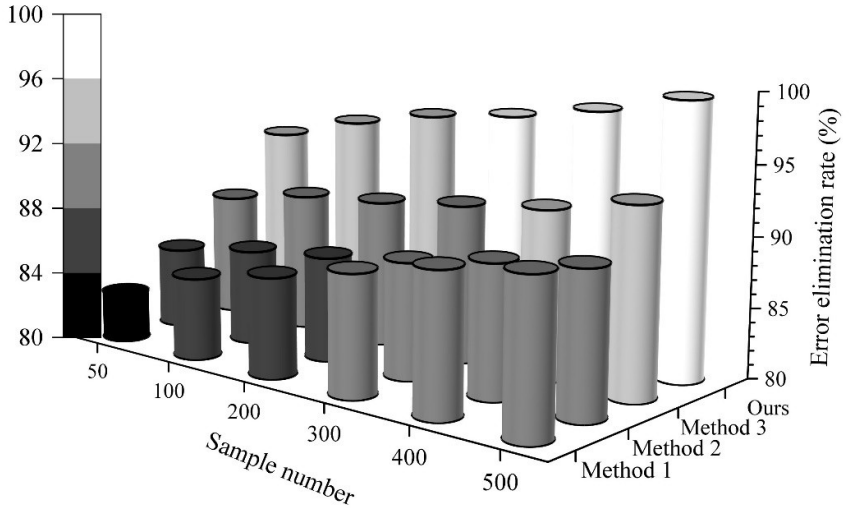


**Fig. 8.** The semantic error elimination rate of all parties is compared

4.2.3. Evaluation of Translation Syntax Optimization. In order to verify the optimization effect of the method proposed in this paper in translation syntax, the BLEU score is chosen as the optimization evaluation index. In this section, under the same experimental environment as above, syntax optimization comparison experiments are carried out with multi-fuzzy semantic automatic judgment, particle swarm optimization algorithm and deep learning error elimination method experimented in the previous section.

Figure 9 shows the comparison results of BLEU values for syntactic optimization. As can be seen from the figure, method 2 has the lowest BLEU value in both "Chinese to English" and "English to Chinese". In this paper, the BLEU value is the highest under the effect of the multi-head attention mechanism. Compared to method 2, the BLEU value on the "Chinese to English" translation is increased by 6.89, and the "English to Chinese" direction is increased by 6.97. It shows that the translated sentences obtained by this method are more suitable for human translation, and the logic is clear and natural, which can form a translation with an easy-to-understand flow field.

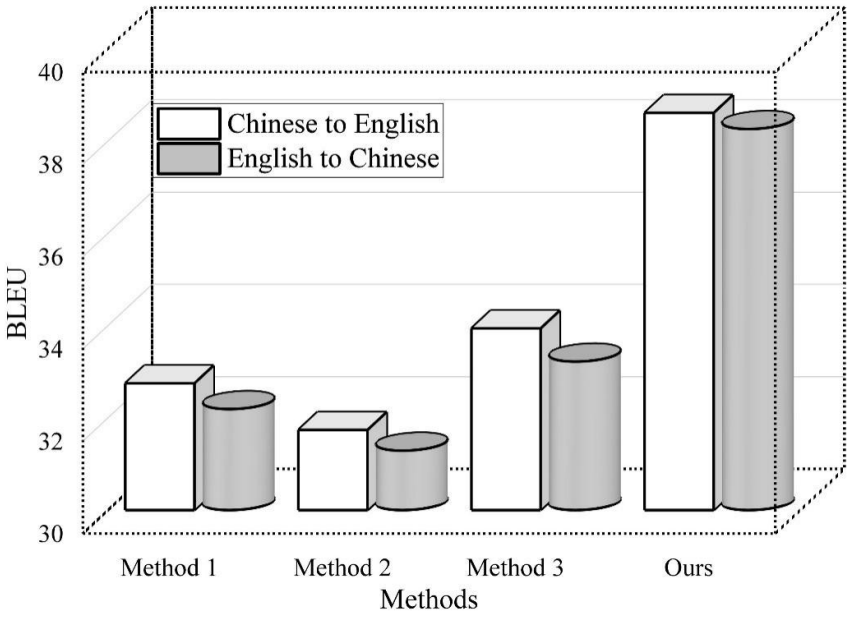


**Fig. 9.** The method of syntactic optimization results in the comparison of BLEU

In order to verify the generalization of the present method, syntactic correctness experiments were conducted on 2000 English long sentences extracted from the IWSLT171, IWSLT152, and IWSLT133 datasets on the topics of news, law, and novels, respectively.

Figure 10 depicts the results of the syntactic correctness comparison of each model in the three datasets. From the figure, it can be seen that the syntactic correct rate results of the methods on English long sentences of different topics in different datasets are deviated. However, the four methods have the best effect on legal long sentences with the correct rate of 84.70% to 96.57%. The effect on the novel dataset is lower, with only 70.74%~88.69% correct rate. The reason is that common legal texts have standardized structure and rigorous content, so their syntactic analysis is more effective. The common fiction text has flexible and changeable language and will use various rhetorical devices, so it leads to a lower effect of its syntactic analysis. And on the long sentences of different topics in the three datasets, the syntactic correctness of this paper's method is shown to be the highest, and in the case of literary topics, this paper's method improves 11.58%~21.42% compared with the comparison method. It indicates that this paper's method pays more attention to the correctness of grammatical elements such as tense, morphology, subject-verb agreement, etc., when translating long sentences, and the syntactic optimization effect is better.

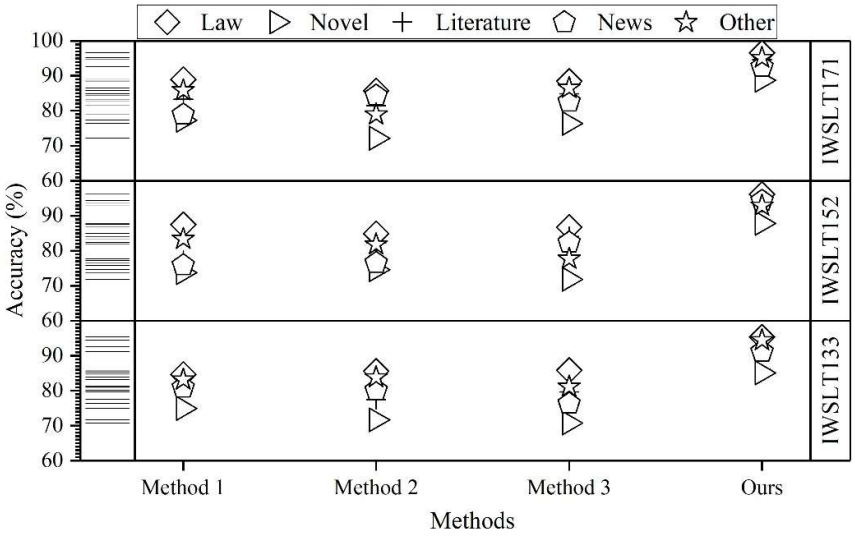

**Fig. 10.** The results of the comparison of the syntax accuracy

## 5. Conclusion

In this paper, we constructed a syntactic optimization and semantic reconstruction model for English translation based on machine learning, and the model takes the EM algorithm as the core, and the training of the Transformer model as an auxiliary task. In the process of corpus generation, the joint learning model is used for parameter optimization. Weights are assigned to the generated corpus through pre-training of word vectors and iterative training of EM based on the Transformer model. After determining the objective function for model training, relevant experimental studies are carried out on a variety of datasets. The experimental results are as follows:

The BLEU score of this paper's model in the training phase converges to about 27.5, which has a good translation effect. The BLEU score reaches 32.86 when the number of layers of the attention mechanism is 1. The feature distribution of the semantic reconstruction of this paper's model is similar to the simulation results in [-1,1], and when the number of samples is 500, the error elimination rate of the semantic reconstruction is improved by 6.56%~10.28% compared with the comparison method, which indicates that this paper's model has strong correlation of translation calibration. The BLEU value of this model in syntactic optimization on is nearly 7 points higher than that of particle swarm optimization algorithm, and it has the highest syntactic accuracy rate, showing excellent syntactic optimization effect of English translation. The model in this paper shows significant advantages in terms of translation quality, fluency and accuracy on both syntactic optimization and semantic reconstruction.

## References

[1] Fan, A., Bhosale, S., Schwenk, H., Ma, Z., El-Kishky, A., Goyal, S., ... & Joulin, A. (2021). Beyond english-centric multilingual machine translation. Journal of Machine Learning Research, 22(107), 1-48.

[2] Tursunovich, R. I. (2022). Linguistic and cultural aspects of literary translation and translation skills. British Journal of Global Ecology and Sustainable Development, 10, 168-173.

[3] Akan, M. F., Karim, M. R., & Chowdhury, A. M. K. (2019). An analysis of Arabic-English translation: Problems and prospects. Advances in Language and Literary Studies, 10(1), 58-65.

[4] Brisset, A., Gill, R., & Gannon, R. (2021). The search for a native language: Translation and cultural identity. In The translation studies reader (pp. 289-319). Routledge.

[5] Malmkjær, K. (2019). Linguistics and the Language of Translation. Edinburgh university press.

[6] Khan, A. B., Mansoor, H. S., & Manzoor, S. (2016). The effectiveness of grammar translation method in teaching and learning of English language at intermediate level. International Journal of Institutional & Industrial Research, 1(1), 22-25.

[7] O'Brien, S., Federici, F., Cadwell, P., Marlowe, J., & Gerber, B. (2018). Language translation during disaster: A comparative analysis of five national approaches. International journal of disaster risk reduction, 31, 627-636.

[8] Cherchata, L., Korol, L., Rubchak, O., Orda, O., & Novytska, D. (2023). Effectiveness of translation transformations in different styles of the english language for teaching written translation. Amazonia Investiga, 12(67), 185-197.

[9] Khan, N. S., Abid, A., & Abid, K. (2020). A novel natural language processing (NLP)–based machine translation model for English to Pakistan sign language translation. Cognitive Computation, 12, 748-765.

[10] Singh, S. P., Kumar, A., Darbari, H., Singh, L., Rastogi, A., & Jain, S. (2017, July). Machine translation using

deep learning: An overview. In 2017 international conference on computer, communications and electronics (comptelix) (pp. 162-167). IEEE.

[11] Ananthanarayana, T., Srivastava, P., Chintha, A., Santha, A., Landy, B., Panaro, J., ... & Nwogu, I. (2021). Deep learning methods for sign language translation. ACM Transactions on Accessible Computing (TACCESS), 14(4), 1-30.

[12] Ali, M. N. Y., Rahman, M. L., Chaki, J., Dey, N., & Santosh, K. C. (2021). Machine translation using deep learning for universal networking language based on their structure. International Journal of Machine Learning and Cybernetics, 12(8), 2365-2376.

[13] Klimova, B., Pikhart, M., Benites, A. D., Lehr, C., & Sanchez-Stockhammer, C. (2023). Neural machine translation in foreign language teaching and learning: a systematic review. Education and Information Technologies, 28(1), 663-682.

[14] Liu, Y., & Zhang, J. (2018). Deep learning in machine translation. Deep learning in natural language processing, 147-183.

[15] Lone, N. A., Giri, K. J., & Bashir, R. (2023). Machine intelligence for language translation from Kashmiri to English. Journal of Information & Knowledge Management, 22(04), 2250074.

[16] Kumar, A., Pratap, A., & Singh, A. K. (2022). Generative adversarial neural machine translation for phonetic languages via reinforcement learning. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(1), 190-199.

[17] Gayam, S. R. (2021). Artificial Intelligence for Natural Language Processing: Techniques for Sentiment Analysis, Language Translation, and Conversational Agents. Journal of Artificial Intelligence Research and Applications, 1(1), 175-216.

[18] Wang, X., Lu, Z., Tu, Z., Li, H., Xiong, D., & Zhang, M. (2017, February). Neural machine translation advised by statistical machine translation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).

[19] Stahlberg, F. (2020). Neural machine translation: A review. Journal of Artificial Intelligence Research, 69, 343-418.

[20] Xu, W., Napoles, C., Pavlick, E., Chen, Q., & Callison-Burch, C. (2016). Optimizing statistical machine translation for text simplification. Transactions of the Association for Computational Linguistics, 4, 401-415.

[21] Koehn, P. (2020). Neural machine translation. Cambridge University Press.

[22] Abbas Mahdavi,Narayanaswamy Balakrishnan & Ahad Jamalizadeh. (2024). EM algorithm for bounded generalized t mixture model with an application to image segmentation. Computational and Applied Mathematics(1),89-89.

[23] Md MijanurRahman,AshikUzzaman,Sadia IslamSami,FatemaKhatun & Md Al AminBhuiyan. (2024). A comprehensive construction of deep neural network‐based encoder–decoder framework for automatic image captioning systems. IET Image Processing(14),4778-4798.

[24] Mingxia Wu,Wei Liu & Shengyang Zheng. (2024). Intelligent Food Safety: A Prediction Model Based on Attention Mechanism and Reinforcement Learning. Applied Artificial Intelligence(1).

[25] Ruxin Gao,Haiquan Jin,Jiahao Chang,Xinyu Li & Qunpo Liu. (2025). A fast gangue detection algorithm based on multi-head self-attention mechanism and anchor frame optimization strategy. International Journal of Coal Preparation and Utilization(1),196-212.

[26] JianBang Liu,Mei Choo Ang,Jun Kit Chaw,Kok Weng Ng & Ah Lian Kor. (2024). Personalized emotion analysis based on fuzzy multi-modal transformer model. Applied Intelligence(3),227-227.

[27] Shuang Wen,Hongru Li & Yinghua Yang. (2025). Inter-temporal dynamic joint learning model considering intra- and inter-day mutable correlations for blood glucose level prediction. Biomedical Signal Processing and Control107204-107204.