International Journal for Housing Science and Its Applications
Publish August 10, 2025. Volume 46, Issue 4 Pages 5474-5482

https://doi.org/10.70517/ijhsa464386

Research on the Optimization of Semantic Consistency in English Translation Systems Using Dynamic Computational Methods

Li Zhang[1,*]
[1] School of Humanities and Design, Henan Open University, Zhengzhou, Henan, 450046, China
Corresponding authors: (e-mail: zhangli2@haou.edu.cn).

Abstract In recent years, neural network technology has been widely used in machine translation, especially in improving translation quality and semantic consistency. In this paper, a translation system optimization model based on a dynamic computational method is proposed. The model adopts the dynamic reconfigurable binarized neural network (DRBNN) computational method to improve the semantic consistency and translation quality of the translation system. Feature interaction layers and grouped sparse regularization terms are introduced into the model to reduce the number of model parameters, and the computational efficiency is improved by quantization methods. In the experiments, the model performs well in several English translation tasks. In particular, in the WMT14 English to German task, the model achieves an accuracy rate of 94.62% and an F1 value of 94.28%; on the AI Challenger dataset, the accuracy rate is 96.24% and the F1 value is 96.05%. The BLEU scores of the model also show high performance under different data volumes, especially in the case of larger data volumes, the BLEU scores are significantly improved. In addition, the model performs well in terms of semantic consistency, reducing the TER values from 2.42 to 10.14 compared to the traditional methods. The experimental results prove that the dynamic computing method proposed in this paper effectively improves the overall performance and semantic consistency of the translation system.

Index Terms dynamic computation method, binarized neural network, feature interaction layer, semantic consistency, translation system, BLEU

I. Introduction

English corpus is a computer-based online encyclopedia of knowledge about English [1]. As a large-scale electronic text database, corpus is undoubtedly an important supply source of English translation resources [2]. English corpus-based translation technology mainly relies on a large number of English language samples to analyze the linguistic rules and features in them in order to achieve more accurate and fluent translation [3], [4]. Although the construction and application of English corpus requires a large amount of work, it can effectively improve the quality of translation and can reduce the time and cost of translation [5]. In English translation, English corpus can help translators quickly find more example sentences of English-Chinese translation in related fields, so as to improve their translation level [6], [7].

In the process of translation, the most important point is that the translation must be faithful to the original text, which is the consensus of almost all translators, and the principle of faithfulness to the original text requires the translator to try to convey all kinds of meanings of the original text, especially the most important ones [8]-[10]. And semantic consistency, as one of the important indexes for evaluating the quality of machine translation, often leads to semantic inconsistency in the translation results due to the differences in the grammatical structures and expressions of the two languages [11]-[13]. Therefore, improving semantic consistency is of great significance for improving the quality of English translation [14].

In this study, an improved translation system incorporating a dynamic computational approach is proposed. The system is based on Dynamic Reconfigurable Binarized Neural Networks (DRBNN), and its core innovation lies in the introduction of a feature interaction layer and a low-bit quantization method to optimize the computational performance of the neural network model. By using grouped sparse regularization terms, the complexity of the model is further reduced, and the overall efficiency and accuracy of the translation system is improved. Specifically, the bottlenecks of the current translation system are first analyzed, and an optimization scheme incorporating dynamic computational methods is proposed. Then, by designing a new neural network structure, a binarized neural network and a feature interaction layer are specifically introduced to strengthen the representation ability of the model and improve the computational efficiency. In terms of experimental design, several public datasets, including WMT14, AI Challenger and IWSLT, are selected for model training and evaluation in this paper. The experimental results show that the proposed method demonstrates strong performance in several translation tasks, especially in translation quality and semantic consistency.

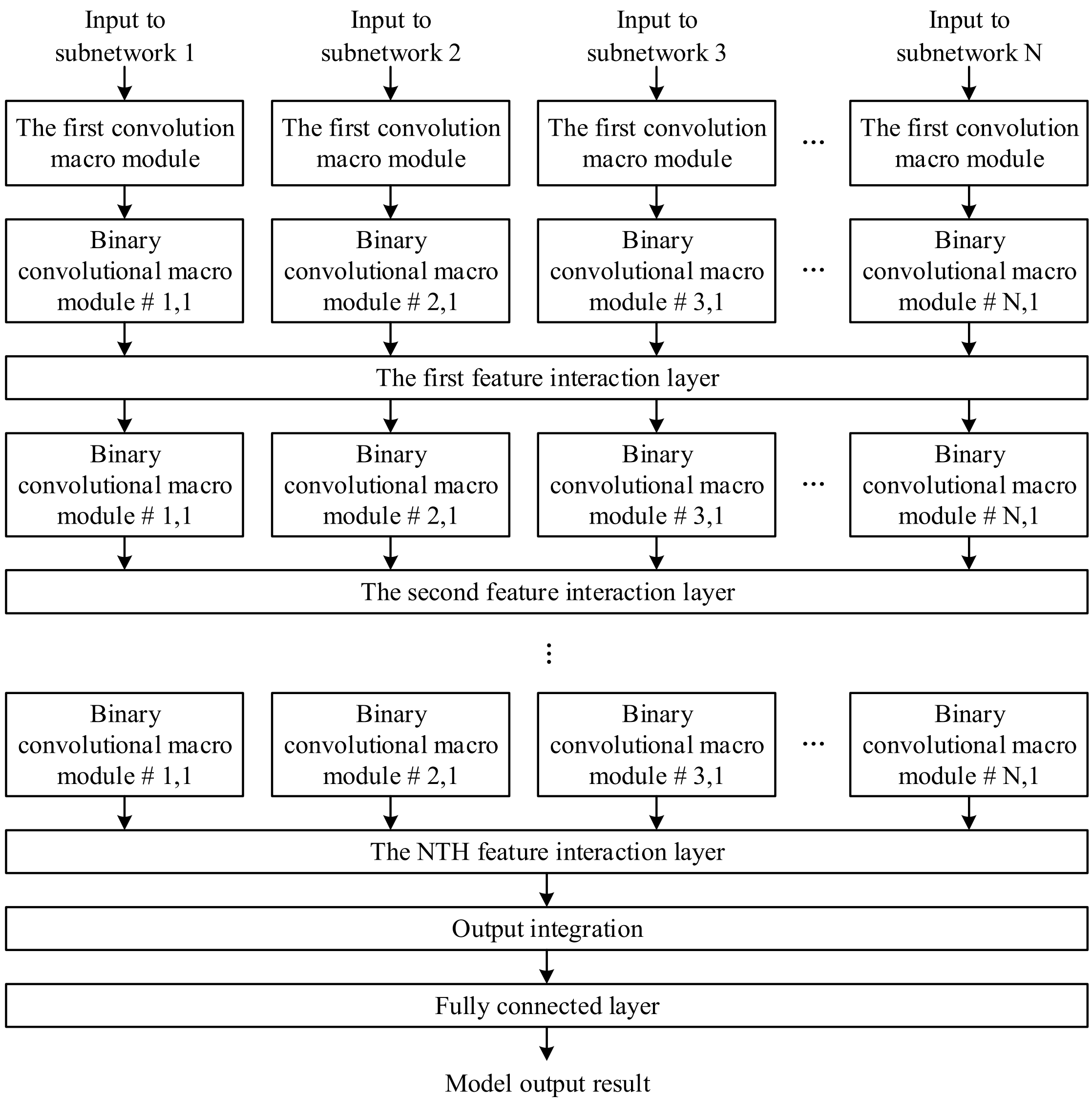

Figure 1: Schematic diagram of the overall model structure of DRBNN

II. Dynamic reconfigurable binarized neural network computational methods

II. A.Binarized Neural Network Computational Methods

In this section, the design and optimization of the binarized neural network computation method is unfolded using BiReal-Net as the benchmark model, and a dynamically reconfigurable binarized neural network computation method (DRBNN) is proposed to enhance the representation capability of the binarized neural network model with the flexibility of dynamically reconfigurable parallelism.

Inspired by the way of information interaction between neurons in adjacent layers of the deep neural network model, DRBNN introduces a structure capable of accomplishing the feature information interaction between different sub-networks, the Feature Interaction Layer (FILayer), using BiReal-Net as a sub-network. According to the feature interaction layer computation rule that the input feature map of binary convolution macro module $\# j,k+1$ is the channel-level weighted sum of the output feature maps of binary convolution macro module $\# j,k$ plus the output feature maps of binary convolution macromodules of other sub-networks at level k, the input feature map $X^{(j,k+1)}$ of binary convolution macro module $\# j,k+1$ can be expressed as follows by Eq:

$$X^{(j,k+1)} = Y^{(j,k)} + \sum_{i}^{i\neq j} \varphi^{(i,j,k)} \circ Y^{(i,k)} \tag{1}$$

where $Y^{(i,k)}$ denotes the output feature map of the binary convolutional macromodule $\#i,k$, $\varphi^{(i,j,k)}$ denotes the weight vectors of the connections in the kth level of the feature interaction layer pointing from the sub-network i to the sub-network j, and $\circ$ denotes the computation of multiplication channel by channel.

More specifically using c to denote the number of channels of the feature map, the feature map $X^{(j,k+1)}$ for the tth channel in $X^{(j,k+1)}$ can be expressed as the following equation:

$$X_t^{(j,k+1)} = Y_t^{(j,k)} + \sum_{i}^{izj} \varphi_t^{(i,j,k)} Y_t^{(i,k)} \tag{2}$$

where $Y_i^{(i,k)}$ denotes the tth sub-feature map of the output feature map of the binary convolutional macromodule $\#i,k$ and $\varphi_i^{(i,j,k)}$ denotes the tth parameter in the connection weight vector $\varphi^{(i,j,k)}$.

Based on the above model structure optimization design, the overall model structure of DRBNN integrating N binarized sub-networks is shown in Fig. 1. Where L is the number of feature interaction layers of the model, i.e., the number of binarized convolutional macromodules contained in each subnetwork, the output integration operation splices and integrates the N feature maps output from the last feature interaction layer in accordance with the dimensions of the channels, so as to obtain the feature maps containing the feature information of all the subnetworks, which are inputted into the fully connected layer for the computation of the model output results.

II. B.Feature Interaction Layer Quantization Method

Denoting the number of bits to be quantized by Q, the weight vector $\varphi^{(i,j,k)}$ connected in the feature interaction layer is first constrained to [-1, +1] by the Tanh function, and then the values in the weight vector are constrained to the interval of [0, 1] to facilitate the input to the quantization function for the low bit quantization by the following formula:

$$\varphi_{restrain}^{(i,j,k)} = \frac{\mathrm{Tanh}(\varphi^{(i,j,k)})}{2\max|\mathrm{Tanh}(\varphi^{(i,j,k)})|} + \frac{1}{2} \tag{3}$$

where $\varphi_{restrain}^{(i,j,k)}$ denotes the weight vector after constraints, and max denotes the maximum value of the parameter in the sought vector. By feeding this vector into a Q-bit quantization function, the floating-point representation of the parameter is quantized to a Q-bit fixed-point number as follows:

$$\varphi_Q^{(i,j,k)} = quantize_Q(\varphi_{restrain}^{(i,j,k)}) = \frac{1}{2^Q - 1} round((2^Q - 1)\varphi_{restrain}^{(i,j,k)}) \tag{4}$$

where round means to round each parameter in the vector separately. This results in a Q-bit fixed-point representation of the weight vector $\varphi_Q^{(i,j,k)}$, which realizes the low-bit quantization of the feature interaction layer. In order to make the model completely free of floating-point parameters, based on the above quantization method of the feature interaction layer, the first convolutional layer of each sub-network and the fully-connected layer of the model are further quantized to reduce the parameter size of the model.

II. C.Grouping sparse regularization terms

In this subsection, an L21-paradigm based regularization term (RT) is proposed to accomplish the weight update of the feature interaction layer with sparse grouping. First, the L2 paradigm regularization term for a single connection weight vector in the kth feature interaction layer is expressed as follows:

$$\|\varphi^{(i,j,k)}\|_2 = \sqrt{\sum_{t=1}^{c} (\varphi_t^{(i,j,k)})^2} \tag{5}$$

where $\varphi_i^{(i,j,k)}$ denotes the parameters in the weight vector, i and j are the indices of the sub-networks in the model, t is the index of the vector channel, and c is the number of channels in the vector, respectively.

Then using Φ to denote the set of weight vectors of all feature interaction layers all connected in the DRBNN, the group sparse regularization term based on the L21 paradigm is represented as follows:

$$\|\Phi\|_{2,1} = \sum_{k=1}^{L} \sum_{i,j}^{i\neq j} \|\Phi^{(i,j,k)}\|_2 \tag{6}$$

where i and j denote the indexes of the subnetworks, k denotes the index of the layer, and L denotes the number of feature interaction layers in the model. Taking the weight vectors of all connections in a single feature interaction layer as a subgroup, this regularization term sums the L2 paradigms of all connection weight vectors in each subgroup as the result of the regularization term computation.

II. D.Loss Functions and Training Methods

Based on the grouped sparse regularization terms of the feature interaction layer mentioned above, and denoting the original loss function of the model based on the cross-entropy (CE) function by L_{CE}, the loss function of the DRBNN can be expressed as:

$$\mathrm{L} = \mathrm{L}_{CE}(\hat{Y}, Y^{g}) + \rho \|\Phi\|_{2,1} \tag{7}$$

where $\hat{Y}$ and Y^{s} denote the predicted labels and true labels of the DRBNN output, respectively, L_{CE} calculates the loss of both and optimizes the parameters of the basic hierarchical structure in the model by minimizing the first loss term during the training process. The ρ, on the other hand, is a hyperparameter used to control the sparseness of the grouping of feature interaction layer connections; the larger the ρ, the sparser the connections of the feature interaction layer in the model. The weight vector of the feature interaction layer introduced in the model is optimized by minimizing the second loss term during the training process.

During the training process of DRBNN, since the feature interaction layer introduces more gradient flow paths for the backpropagation of the model, according to the chain derivation rule, the gradient flowing through the t-channel of the k-convolutional layer of the j-subnetwork in the model outputting the feature map $Y^{(j,k)}$ can be calculated by the following formula:

$$\frac{\partial \mathrm{L}}{\partial y_t^{(j,k)}} = \frac{\partial \mathrm{L}}{\partial x_t^{(j,k+1)}} \frac{\partial x_t^{(j,k+1)}}{\partial y_t^{(j,k)}} + \sum_{i}^{i \neq j} \frac{\partial \mathrm{L}}{\partial x_t^{(i,k+1)}} \frac{\partial x_t^{(i,k+1)}}{\partial y_t^{(j,k)}} \tag{8}$$

where $y_i^{(j,k)}$ is the parameter of the feature map $Y^{(j,k)}$ t rd channel subfeature map $Y_t^{(j,k)}$ and $x_i^{(j,k+1)}$ is the parameter of the jth subnetwork $k+1$th convolutional layer input feature map $X^{(j,k+1)}$ tth channel subfeature map $X_i^{(j,k+1)}$. Unlike single-branch neural networks and BENNs, the gradient of the output feature map parameter $y_t^{(j,k)}$ in the DRBNN model consists of not only the backpropagation gradient of the self-subnetwork, but also the backpropagation gradients of the other binary subnetworks, and more specifically the formula for the feature interaction layer. The backpropagation formula for the parameters in the model can be rewritten as:

$$\frac{\partial \mathrm{L}}{\partial y_t^{(j,k)}} = \frac{\partial \mathrm{L}}{\partial x_t^{(j,k+1)}} + \sum_{i}^{i \neq j} \varphi_t^{(j,i,k)} \frac{\partial \mathrm{L}}{\partial x_t^{(i,k+1)}} \tag{9}$$

where $\varphi_i^{(j,i,k)}$ denotes the tth parameter of the weight vector $\varphi^{(j,i,k)}$ in the feature interaction layer.

In addition, DRBNN optimally updates the weight vector $\varphi^{(i,j,k)}$ in the feature interaction layer in the backpropagation, and the gradient of each channel parameter $\varphi_t^{(i,j,k)}$ in this vector is computed as shown below:

$$\frac{\partial \mathrm{L}}{\partial \varphi_t^{(i,j,k)}} = \sum_{x_t^{(j,k+1)}} \frac{\partial \mathrm{L}}{\partial x_t^{(j,k+1)}} \frac{\partial x_t^{(j,k+1)}}{\partial \varphi_t^{(i,j,k)}} \tag{10}$$

where the cumulative symbol indicates that this formula acts on all parameters of the tth channel subfeature map $X^{j,k+1}$ of the feature map $X^{j,k+1}$.

III. Optimization of the translation system

The topic of this paper is the study of translation system optimization incorporating dynamic computational methods. Therefore, this section builds a complete integrated translation system based on document configuration using the above dynamic reconfigurable binarized neural network computation method. The system starts from text corpus processing and integrates word vector transformation, model training, translation and BLEU value evaluation. During the period, all eligible models during the whole training process were evaluated and the optimal model was dynamically retained.

The dynamic model evaluation during training allowed us to observe the training process of the whole translation system and understand the trend of model training. Since our training model save file is relatively large, the dynamic preservation of the optimal model can save us a lot of storage space.The embedding of BLEU value evaluation makes it unnecessary for us to copy the translation results from the NMT to the SMT system for evaluation. Instead, we can evaluate the results directly. We only need to store the English parallel corpus in a fixed directory structure, and then run a shell script with a file configuration to train, translate and evaluate the model and get the final results. The architecture of the whole translation system is shown in Figure 2.

First, the parallel training corpus is put into the word vector module for word vector training. After deriving the word vectors, all sentence pairs in the training corpus are loaded in the form of word vectors, and batch-size parallel sentence pairs are input to the translation system for training each time. Before proceeding to the next Epoch, a shuffle operation is performed on the whole corpus, so that the set of sentence pairs in each iteration of the next training is different from that of the last one, which better reflects the linguistic features of the whole training corpus.

At the beginning of the training process, after each saveFreq iteration, the error value is checked to see if it is within the pre-set error range. If it is, a model will be generated temporarily, and the model will be utilized for translation and evaluation, and the translation and evaluation results will be saved. At the first saving of the model, the model will be saved directly. Otherwise, it will determine whether the BLEU value of the translation result of the current model is higher than the BLEU value of the translation result of the best existing model. If it is higher, then the original model will be deleted and the current model will be kept, otherwise the current model will be deleted.

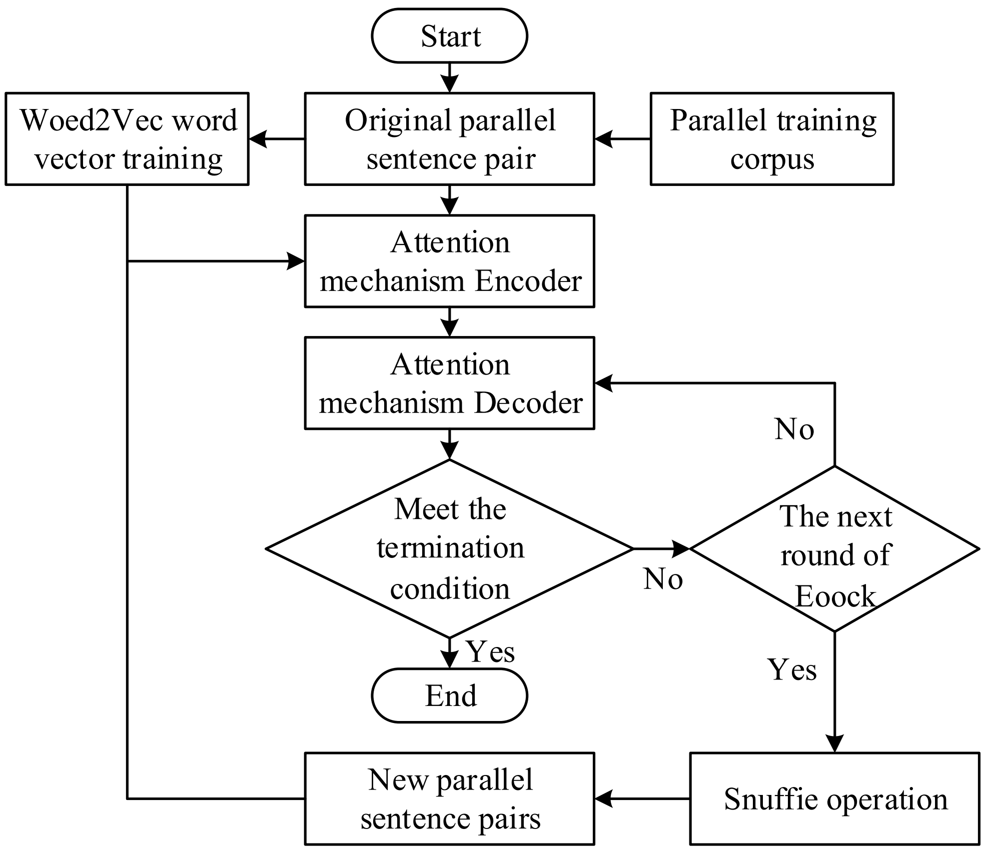

Figure 2: Training process of the attention neural network translation system

IV. Experimental assessment setup

IV. A. English corpus selection

For the improved translation system incorporating the dynamic computing method proposed in this paper, the effectiveness of the method is evaluated by conducting experiments on publicly available benchmark datasets. The experiments in this study use four translation task datasets, namely, the English-to-German translation task dataset of the 2014 Workshop on International Machine Translation Competition (WMT14), the Chinese-to-English translation task dataset of the AI Challenger 2018, the German-to-English translation task dataset of the 2014 International Spoken Language German-to-English translation task dataset from the 2014 International Word of Spoken Language Evaluation Competition (IWSLT14), and the Chinese-to-English dataset from IWSLT15.

The WMT14 and AI Challenger datasets are used to test the effectiveness of the method of this study in the case of large-scale corpus, and the IWSLT14 and IWSLT15 datasets are used to test the effectiveness of the method in the case of small-scale corpus. Among them, the WMT14 English to German training set totaled about 4.5 million sentence pairs, and “AI Challenger 2018” contains 12 million training data.The IWSLT14 German to English dataset is from the machine translation track of the IWSLT 2014 evaluation competition, and the bilingual training data totaled about 160K sentence pairs, and the IWSLT15 Chinese-to-English dataset totaled 21K bilingual sentence pairs.

IV. B. Methods of comparison

This study compares in detail three representative methods that have been developed: sentence-level neural machine translation methods, chapter-level neural translation methods, and dialog translation methods:

(1) Transformer: the standard Transformer architecture, which is trained only on WMT sentence data without being fine-tuned by dialog translation data.

(2) Transformer+FT: a model that is fine-tuned with dialog translation data after pre-training on a sentence-level translation corpus.

(3) Doc-Transformer+FT: A Transformer-based chapter-level translation model that incorporates bilingual dialog history by sharing the first layer encoder.

(4) Dia-Transformer+FT: This model uses additional RNN encoders to incorporate bilingual dialog history, but in this paper, the approach is reimplemented based on the Transformer architecture and uses an additional Transformer layer to introduce dialog history.

(5) V-Transformer+FT: Transformer-based variational translation model. For fair comparison, it also incorporates bilingual contextual information by sharing the first layer of encoder.

IV. C. Evaluation indicators

In order to make a fair comparison, two indexes, BLEU (the higher the better) and TER (the lower the better), were used, and a statistical significance test was conducted.The calculation of BLEU, TER and statistical significance test is illustrated below.

BLEU (BiLingual Evaluation Understudy) is the degree of similarity between computerized and human translations based on the multivariate grammatical matching algorithm, so it is also called multivariate grammatical accuracy (N-gram precision). The higher the similarity, the higher the quality of the translation, and its value is between 0 and 1. Formally, BLEU is calculated as shown below:

$$P_n = \frac{count_{hit}}{count_{output}} \tag{11}$$

where $count_{hit}$ denotes the number of hits of the multivariate grammar in the machine translation in the human translation, and $count_{output}$ denotes the number of multivariate grammars in the machine translation. The multivariate grammar matching accuracy is calculated in the following form:

$$P_{avg} = \exp\left(\sum_{n=1}^{N} w_n \cdot \log P_n\right) \tag{12}$$

where n denotes the maximum multivariate grammar length, and n generally takes values from 1 to 4. It can be seen that the result is a weighted average of multiple multivariate grammars. However, sometimes using only P_{avg} may cause the model to tend to output shorter sentences. So a length penalty factor is often added to the application:

$$BP = \begin{cases} 1, & c > r \\ \exp(1 - \frac{r}{c}), & c \le r \end{cases} \tag{13}$$

where c denotes the number of words in the machine translation and r denotes the number of words in the human translation. Then the final BLEU is calculated as:

$$BLEU = BP \cdot P_{avg} \tag{14}$$

TER (Translation Edit Rate) is a method of evaluating machine-translated translations by calculating the amount of post-translation editing work (edit distance). The more edits, the larger the distance, and the lower the similarity between translations. On the contrary, the smaller the distance, the easier it is to rewrite one sentence into another and the higher the similarity between translations. Generally speaking, there are four kinds of editing operations: shift, delete, replace and add. Then edit distance is the distance obtained by calculating the number of operations of delete, replace and add. In calculating the distance, all operations have a cost of 1. A score is given in the form of an error rate:

$$TER = \frac{edit(c,r)}{\ell} \tag{15}$$

where $edit(c,r)$ denotes the distance between the machine translation c and the human translation r, and ℓ is the normalization factor, usually the length of the human translation. In general, the calculation will prioritize the shift operation before calculating the edit distance. Until the "shift operation" cannot reduce the editing distance, the editing distance and the number of times generated by the "shift operation" will be added up to get the final TER result.

V. Research on semantic consistency of translation systems based on dynamic computing methods

V. A. Improvement model English translation performance test

The translation system uses a MYSQL database and a knowledge base as the data management system. The Common Word Dictionary and WordNet Semantic Dictionary were used as data tables. The selected test cases are as follows:

Test case 1: English to German example sentences in the WMT14 corpus.

Test case 2: Chinese to English translation task in the "AI Challenger 2018" challenge.

Test case 3: German to English corpus test of the 2014 International Spoken Language Machine Translation Assessment Competition.

Test case 4: IWSLT15 Chinese to English dataset.

The English corpus test results of the improved translation system are shown in Table 1. From the experimental results obtained, it can be seen that the translation system optimized with the algorithm of this paper to solve the English translation problem can obtain a high accuracy rate. Among them, the accuracy rate, call rate and F1 value of test case 4 are lower than the other three, which may be due to the fact that the total number of sentence pairs in case 4 is less, only 21K. And it is mainly interwoven with narrative and expository texts, with more variations in tense and some errors. And the accuracy, call rate and F1 value in other test cases are more than 90%, especially the accuracy, call rate and F1 value of test case 2 are 96.24%, 95.87% and 96.05% respectively. The method of this paper is based on establishing a corpus and determining the relevant semantic extraction rules and dependencies, which can effectively differentiate utterances and words lexically, syntactically, and tense multifaceted, and further enhance the efficiency of the translation system.

Table 1: Improve the English language material test results of the translation system

	Sentence total number	Failure number	Recall (%)	Precision (%)	F1 (%)
Test case 1	4500K	221761	93.94	94.62	94.28
Test case 2	12000K	435617	96.24	95.87	96.05
Test case 3	160K	11375	91.93	90.64	91.28
Test case 4	21K	2468	87.89	86.55	87.21

In order to further validate the ability of the proposed algorithm to be applied in each English corpus, English translation performance tests were conducted in the above dataset. The experimental code was deployed in GPU and the TensorFlow framework was utilized for code testing. The number of different English training corpus was chosen to be 1000, 5000 and 10000 respectively, and the average BLEU and TER values were taken after running the training dataset for 5 times. The English translation test results are shown in Table 2.

From the test data in the table, it can be seen that with the increase in the amount of data, the BLEU value and TER value of the translation system show an increasing and decreasing trend, respectively. It shows that the larger the data volume of the English corpus, the better the performance of the optimized translation system of this paper's algorithm. When the data volume is 1000 entries, the BLEU values of the translation system on the WMT14, AI Challenger, IWSLT14, IWSLT15 datasets are 29.87, 31.71, 32.13, 30.77, and the TER values are 19.62, 20.59, 22.07, 20.24, respectively.The translation system under this paper's algorithm performs better on the all number of English corpora performs best.

Table 2: English translation test results

Index	Date set	1000	5000	10000
BLEU	WMT14	29.87	31.05	32.09
	AI Challenger	31.71	33.26	34.12
	IWSLT14	32.13	33.25	34.55
	IWSLT15	30.77	32.16	33.85
TER	WMT14	19.62	18.91	16.83
	AI Challenger	20.59	20.10	19.71
	IWSLT14	22.07	21.91	19.22
	IWSLT15	20.24	18.84	17.53

V. B. Translation complexity and semantic consistency analysis

This section analyzes the model parameters and training speed on the WMT14, AI Challenger, IWSLT14, and IWSLT15 datasets, and compares Transformer, Transformer+FT, Doc-Transformer+FT, Dia-Transformer+FT, and the dynamics of the proposed dynamic computational methods in terms of time and space complexity.

Table 3 shows the parameter and training speed test results of each method on different datasets. The results show that the number of parameters of this paper's method is much higher than that of the comparison method in different datasets. The number of parameters of this paper's method ranges from 54 to 68 due to the fact that the method introduces steps such as feature interaction and regularization. This also improves the training time of the model, which is distributed between 550 and 760. Despite the increase in training time, this also significantly improves translation quality and semantic consistency, especially when dealing with long documents.

Table 3: The parameters and training speed test results of various data sets are tested

Index	Method	WMT14	AI Challenger	IWSLT14	IWSLT15
Param	Transformer	42.18	48.98	49.06	39.24
	Transformer+FT	44.21	51.77	52.78	41.95
	Doc-Transformer+FT	47.4	54.06	54.94	43.68
	Dia-Transformer+FT	51.02	59.34	57.24	49.96
	Ours	57.87	67.79	63.46	54.91
Speed	Transformer	597	674	501	555
	Transformer+FT	545	626	452	506
	Doc-Transformer+FT	586	703	503	556
	Dia-Transformer+FT	516	603	459	495
	Ours	637	751	545	609

Figures 3 and 4 show the results of BLEU and TER scores for this paper's method and the comparison method on the 4 English corpora, respectively. The experimental results show that the model in this paper achieves higher BLEU scores in the vast majority of cases on the 4 datasets, with BLEU scores ranging from 32 to 36. It proves the effectiveness of the proposed dynamic computing method in improving machine translation. In addition, the model in this paper reduces the TER on all test sets by 2.42 to 10.14, which is significantly better than the comparison method. The experimental results demonstrate the effectiveness of the dynamic computational approach in improving the semantic consistency of machine translation systems.

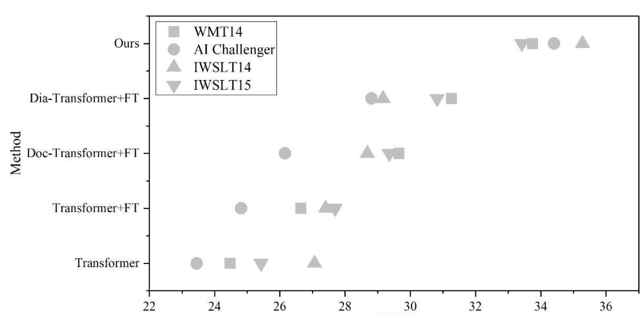

Figure 3: The method and comparison method of the BLEU score in the method

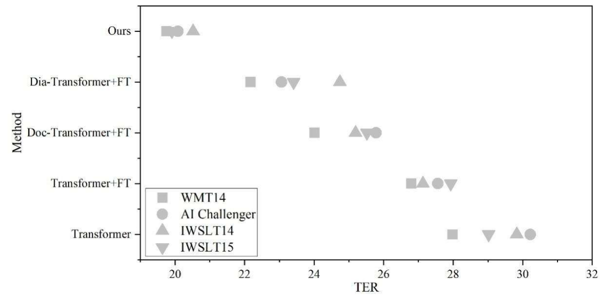

Figure 4: The method and comparison method of the TER score in the method

VI. Conclusion

The translation system based on the dynamic computing method proposed in this paper performs well in several translation tasks, especially in improving the semantic consistency of the translation system with remarkable results. The experimental results show that in the WMT14 English to German translation task, the system with this paper's method achieves an accuracy rate of 94.62% and an F1 value of 94.28%; in the AI Challenger dataset, the accuracy rate is 96.24% and the F1 value is 96.05%. With the dynamic model evaluation and preservation mechanism, the translation system is able to evaluate the model performance in real time during the training process, thus maintaining the optimal model. In addition, with the increase of data volume, the BLEU score of the translation system shows an upward trend, which proves the advantage of this paper's method in large-scale data processing. In terms of semantic consistency, the TER value of this paper's model is significantly lower than that of the comparison method, with a maximum reduction of 10.14, which further verifies the effectiveness of the dynamic computation method in improving the quality of machine translation. Therefore, the method in this paper not only improves the accuracy of the translation system, but also significantly enhances its semantic consistency in complex contexts, which provides new ideas and directions for the optimization of translation systems in the future.

Funding

This work was supported by 2024 Henan Province Philosophy and Social Science Planning Project: Research on the Dissemination Path of National Intangible Cultural Heritage in the Henan Section of the Yellow River Basin Based on Parallel Corpora (2024CYY030) and by Science and Technology Key Project of Henan Province: Research on the Translation and Dissemination Path of National Intangible Cultural Heritage in the Henan Section of the Yellow River Basin (242102320298) .

References

[1] Greenbaum, S. (2014). The development of the International Corpus of English. In English corpus linguistics (pp. 83-91). Routledge.
[2] Meyer, C. F. (2023). English corpus linguistics: An introduction. Cambridge University Press.
[3] Lindquist, H. (2018). Corpus linguistics and the description of English. Edinburgh University Press.
[4] Tianmei, B. (2020, October). Research on English vocabulary translation technology based on dynamic bilingual corpus. In Proceedings of the 2020 International Conference on Computers, Information Processing and Advanced Education (pp. 288-291).
[5] Cong, Y. (2021, September). Research on the Construction of English Translation Corpus Based on Network Electronic Technology. In 2021 4th International Conference on Information Systems and Computer Aided Education (pp. 1152-1156).
[6] Long, J. (2022). Application of Artificial Intelligence (AI) technology in Chinese English translation system corpus. Journal of Artificial Intelligence Practice, 5(3), 8-13.
[7] Hrytsiv, N., Shestakevych, T., & Shyyka, J. (2021, June). Corpus Technologies in Translation Studies: Fiction as Document. In MoMLeT+ DS (pp. 327-343).
[8] Du, Y., Ma, Y. F., Xie, Z., & Li, M. (2023, December). Beyond Lexical Consistency: Preserving Semantic Consistency for Program Translation. In 2023 IEEE International Conference on Data Mining (ICDM) (pp. 91-100). IEEE.
[9] Qin, P., Tan, W., Guo, J., Shen, B., & Tang, Q. (2021). Achieving semantic consistency for multilingual sentence representation using an explainable machine natural language parser (mparser). Applied Sciences, 11(24), 11699.
[10] Jung, S., Choi, W. S., Choi, S., & Zhang, B. T. (2022, May). Language-agnostic semantic consistent text-to-image generation. In Proceedings of the Workshop on Multilingual Multimodal Learning (pp. 1-5).
[11] Qin, T. (2025, January). Semantic Understanding in English Translation Big Data Model Based on Attention Mechanism. In 2025 Asia-Europe Conference on Cybersecurity, Internet of Things and Soft Computing (CITSC) (pp. 738-743). IEEE.
[12] Zhang, X. E. (2021). A study of cultural context in Chinese-English translation. Region-Educational Research and Reviews, 3(2), 11-14.
[13] Sevarakhon, K. (2025). SEMANTIC ANALYSIS IN TRANSLATION. PEDAGOGICAL SCIENCES AND TEACHING METHODS, 4(45), 251-254.
[14] Cao, Y. (2025). The Automatic Scoring Method of English Translation for College Students Considering Latent Semantics. Journal of Information & Knowledge Management, 2550019.