A Lightweight Translation Architecture for Embedded Devices via Multilingual BERT Distillation and Quantization

Wang Yiqing
Institute of Modern Media Technology and Art, Shanghai Publishing and Printing College, Shanghai Yangpu 200093, China
E-mail: yiqingwangg@outlook.com

Keywords: multilingual BERT, knowledge distillation, embedded translation, model compression, cross-language semantic alignment

Received: July 3, 2025

In order to solve the problems of difficult deployment and high computational overhead of multilingual BERT models in embedded devices, this paper proposes a lightweight translation model integrating knowledge distillation technology. The teacher-student knowledge transfer mechanism uses a combination of layer-wise and attention-based distillation strategies, along with optimization techniques such as pruning and 8-bit quantization. BLEU scores were evaluated by comparing the student model against the teacher model and baseline systems, showing competitive translation quality. By constructing a knowledge transfer mechanism between the teacher model and the student model, combined with optimization strategies such as pruning and quantification, the synergistic improvement of model compression and reasoning speed is achieved. The experimental results show that the model has a BLEU value of 28.7 in the WMT-14 English-German task, which is only 1.4 points lower than the teacher model, and retains about 95.3% of the translation quality; The accuracy rate on the XNLI cross-language reasoning dataset reaches 78.3%, which is only 3.1% lower than the teacher model. On the embedded device Jetson Nano, the inference latency of the distilled student model dropped from 1280ms of the teacher model to 195ms, achieved through optimization techniques such as hardware acceleration, resulting in a boost speed of approximately 6.56 times. The proposed model achieves an exceptional compression, reducing the model size from the original 4.2 GB to just 650MB, a reduction of 86.4%. This size reduction is achieved with minimal quality loss, with a BLEU drop of no more than 1.8, ensuring that the compressed model retains most of the performance of the original model. The compressed model has been successfully deployed on edge platforms, including the Raspberry Pi 4B, making it highly suitable for resource-constrained environments. In terms of parameter quantity, the original mBERT has about 1100M parameters, and the distillation model is 350M. After combining pruning and 8-bit quantization, only 137.5 M is left, and the inference speed is increased to 8 times that of the original model. In addition, by introducing the attention distillation mechanism in low-resource scenarios, the model's BLEU score improves by 4.2%, demonstrating the mechanism's effectiveness in enhancing semantic alignment for languages with limited resources. The power consumption test shows that the average power consumption of the student model is 4-6W, which is about 35% lower than that of the original model. Additionally, the memory footprint of the student model on the Raspberry Pi 4B is measured at 320MB during inference, a significant reduction compared to the 1.5GB required by the original mBERT model. These optimizations not only improve translation efficiency and energy efficiency but also provide a highly feasible solution for future deployment of multilingual smart devices.

Povzetek: Predlagani model z destilacijo znanja učinkovito zmanjša velikost in porabo virov mBERT-a ob minimalni izgubi kakovosti ter tako omogoča hitro in energijsko učinkovito rabo na vgrajenih napravah.

1 Introduction

With the acceleration of globalization and the increasing demand for cross-language communication, machine translation technology has become one of the important research directions in Natural Language Processing (NLP) [1]. In recent years, pre-trained language models based on deep learning have achieved remarkable results in multiple NLP tasks, especially in semantic understanding and context modeling [2]. As a powerful cross-language model, Multilingual BERT has shown excellent performance in machine translation tasks in multiple languages. However, despite its outstanding performance on large computing platforms, its application on resource-constrained embedded devices is limited due to its huge number of parameters and computational requirements [3]. Therefore, how to reduce the computational complexity and resource consumption of the model while ensuring the translation

quality has become a key problem in the current application of machine translation technology.

To overcome this challenge, Knowledge Distillation technology has been proposed as an effective model compression method [4]. Through knowledge distillation, the knowledge of a large and complex teacher model can be transferred to a smaller and lighter student model, thus greatly reducing the computational overhead while maintaining the model's performance [5]. Combining the advantages of multilingual BERT and knowledge distillation, it is expected to realize an efficient translation model on embedded devices. Through this method, multilingual BERT's powerful language modeling capabilities can be combined with the compression effect of knowledge distillation, providing a novel solution for efficient translation on embedded devices [6].

However, although knowledge distillation has shown great potential in model compression, effectively preserving translation quality and ensuring that the model can respond in real time under the computing resources and storage limitations of embedded devices is still an urgent problem to be solved. Embedded devices usually have low processing power and memory capacity, and traditional BERT models often cannot be directly applied to these devices [7]. Therefore, this study proposes a lightweight translation model based on the fusion of multilingual BERT and knowledge distillation, which aims to enable the translation system to run on embedded devices by optimizing the distillation process and model design while maintaining high translation accuracy and real-time performance [8].

The contribution of this study lies in combining the powerful language modeling capabilities of multilingual BERT with the model compression technology of knowledge distillation to provide an efficient translation solution for embedded devices [9]. Through this method, we solve the computational and storage problems in translation tasks and improve the accuracy and robustness of cross-language translation. The next work will briefly introduce the combination of multilingual BERT and knowledge distillation, discuss its implementation and optimization strategy on embedded devices, and finally provide an efficient translation model suitable for a resource-constrained environment [10].

2 Theoretical basis and related re search

2.1 BERT and knowledge distillation fusion theory

BERT (Bidirectional Encoder Representations from Transformers) is a pre-trained language model based on the Transformer architecture that excels in natural language processing tasks. Unlike the traditional one-way language model, BERT adopts a two-way context understanding mechanism, enabling it to capture the complex relationship between each word in a sentence and the context, thereby improving performance in various NLP tasks [11]. Through large-scale corpus pre-training, BERT generates high-quality language representations thatthat can understand texts' grammatical and semantic information well. It is widely used in tasks such as question-answering systems, text classification, and named entity recognition [12]. However, the BERT model's huge number of parameters and computational requirements make it difficult to directly apply in environments with limited computational resources, such as embedded devices. Therefore, how to reduce the computational complexity of the model while ensuring the translation quality has become the focus of research.

Knowledge Distillation is an effective model compression method, which enables the student model to maintain high performance while significantly reducing computational overhead and memory footprint by transferring the knowledge of a complex large "teacher" model to a smaller "student" model [13]. The core idea of knowledge distillation is to train the student model to imitate the behavior of the teacher model as much as possible. The student model not only performs supervised learning through labels but also enhances its expressive ability by learning the output of the teacher model. Since teacher models usually have great limitations in model size and computing power, student models can reduce resource consumption and delay while maintaining certain performance, which is very suitable for resource-limited environments such as embedded devices [14].

Combining BERT with knowledge distillation can effectively combine BERT's powerful language modeling capabilities with model compression technology to form a lightweight translation model. In this process, the BERT model acts as the teacher model, and its depth representation and contextual information are passed into a smaller student model [15]. Through knowledge distillation, student models can learn complex language features extracted in BERT and thus run on devices with limited computing resources without significantly sacrificing translation quality. Through this method, the model can run efficiently in embedded devices and maintain high translation accuracy, thus realizing lightweight processing of cross-language translation [16].

However, effectively integrating BERT with knowledge distillation on embedded devices still faces many challenges. Embedded devices often have limited computing power and memory, and direct deployment of BERT models may result in slow or inoperable operation [17]. Therefore, it is necessary to optimize the distillation strategy during the model training process so that the student model is not only compressed in size but also close to the teacher model in accuracy. Specifically, by carefully designing the structure of the student model and adjusting the loss function and training strategy in the distillation process, the computational overhead can be minimized while ensuring performance. This combination of BERT and knowledge distillation can provide an efficient and accurate solution for machine translation in resource-constrained environments [18].

2.2 Current status of lightweight translation models integrating BERT and knowledge distillation

With the continuous development of multilingual machine translation technology, deep learning-based models perform well in various translation tasks, especially the emergence of the BERT (Bidirectional Encoder Representations from Transformers) model, which has greatly promoted the progress in the field of NLP [19]. Through the bidirectional context modeling mechanism, BERT can deeply understand the semantics of each word in a sentence and capture the global context's information, making it excellent in multilingual translation. However, despite the remarkable achievements of BERT in various NLP tasks, its huge number of parameters and high computational requirements make its application in resource-constrained environments such as embedded devices and mobile devices difficult [20]. Therefore, how to combine the powerful capabilities of BERT with the saving of computing resources has become a hot issue in the field of machine translation. The comparison of key indicators of SOTA model is shown in Table 1.

Table 1: Comparison of key indicators of SOTA model

Model	BLEU Score	Model Size (GB)	Inference Time (ms)	Parameters (Million)
Original mBERT	30.1	4.2	1280	1100
Distilled Model	28.7	0.65	195	350
LSTM-based Lightweight Model	25.4	0.5	250	50
TinyBERT	26.3	0.35	180	30
ALBERT	27.5	0.55	220	12

To solve the application problem of BERT in embedded devices, scholars have proposed various model compression and acceleration methods, among which Knowledge Distillation (KD) is considered the most effective [21]. Knowledge distillation transfers its knowledge into a smaller and more efficient "student model" by taking a complex and superior performance "big model"-usually a teacher model. The student model learns the output of the teacher model, thereby acquiring similar abilities to the teacher model while greatly reducing the demand for computing resources. In multilingual translation tasks, using knowledge distillation can effectively reduce the size of the translation model, improve its running efficiency on embedded devices, and reduce latency [22]. Combining the technology of BERT and knowledge distillation, this method can meet the strict limitation of computing resources of embedded devices while ensuring the translation quality in multilingual translation tasks [23].

The lightweight translation model combining BERT with knowledge distillation has achieved initial success in some research. It has been shown that combining the BERT-based teacher model and knowledge distillation can achieve efficient translation performance on embedded devices [24]. For example, through the distillation process, the student model can imitate the efficient reasoning ability of the teacher model in multilingual translation tasks while maintaining high translation quality during the model compression process. However, despite some research progress, how to better maintain the cross-language ability of the multilingual BERT model in the process of knowledge distillation and how to optimize the performance of the student model through more detailed distillation strategies is still one of the difficulties in current research [25]. Especially when dealing with complex language and long text translation, how to balance translation accuracy and computational efficiency is still an important research topic.

To further improve the performance of the lightweight translation model integrating BERT and knowledge distillation, existing research is exploring various improvement strategies. For example, researchers are trying to further optimize the performance of the student model by adjusting the hyperparameters in the distillation process and designing a more flexible student model architecture. In addition, how to make student models achieve better generalization ability among languages is also a current research hotspot for multilingual translation tasks. With the continuous improvement of hardware performance, the lightweight translation model combining BERT and knowledge distillation is expected to be more widely used in embedded devices, mobile devices, and other environments and promote the development of cross-language translation technology to meet more diverse practical needs.

3 Establishment of lightweight tra nslation model for embedded de vices integrating multilingual B ERT and knowledge distillation

3.1 Overall model framework and process

This study presents a lightweight translation model that combines multilingual BERT with KD to address the computational limitations of embedded devices. By transferring knowledge from a large teacher model to a

smaller student model, the approach reduces model size and computational burden, enabling efficient translation on resource-constrained devices. The student model uses 6 Transformer layers and a hidden dimension size of 512, compared to the original BERT's 12 layers and 768 hidden dimensions, significantly lowering computational requirements while maintaining high translation quality. The student models used in this study include the TinyBERT variant with 6 layers and 512 hidden dimensions, and the ALBERT model with 6 layers, where each layer shares parameters across layers. These architectural choices were made to reduce the model size and computation while maintaining competitive translation quality. The formula of the model compression ratio is shown in (1).

$$MCR = \frac{\theta_t}{\theta_s} \times 100\% \quad (1)$$

Where θt represents the total number of parameters of the teacher model. θs denotes the total number of parameters of the student model. The formula of the relationship between inference delay and model complexity is shown in (2).

$$LAT = \beta \cdot \theta_s + \gamma \cdot n_{layers} + \delta \quad (2)$$

Where LAT denotes inference delay. θs represents the parameter quantity of the student's model. nlayers represents the number of Transformer layers of the student model. β,γ,δ represent empirical coefficients. The design of the model framework follows two core ideas: first, using the powerful cross-language representation capabilities of multilingual BERT to improve the quality of multilingual translation; The second is to optimize the computational efficiency of the model through the compression effect of knowledge distillation to ensure that it has sufficient execution speed and real-time performance on embedded devices. Specifically, the framework includes two major modules: the BERT module and the knowledge distillation module. The BERT module is responsible for training and reasoning translation tasks using multilingual BERT to ensure translation quality. The knowledge distillation module compresses BERT knowledge into a smaller student model through knowledge transfer, thereby reducing computational overhead and achieving efficient translation on embedded devices. The framework of a lightweight translation model for embedded devices integrating multilingual BERT and knowledge distillation is shown in Figure 1.

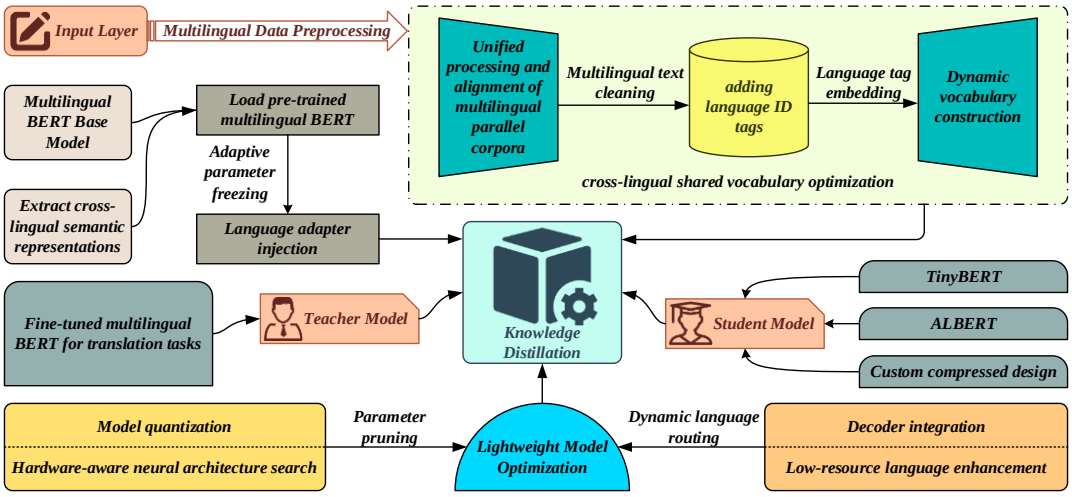

Figure 1: Lightweight translation model framework for embedded devices integrating multilingual BERT and knowledge distillation

In order to improve performance in low resource language scenarios, the model employs several techniques, including synthetic data augmentation, inverse transformation, and zero sample transmission. Synthetic data is generated by interpreting and translating into high resource languages, and then translating into the target language to create additional training samples. Back translation enriches the training set by translating target language sentences into high resource languages and then re translating them back. In addition, zero sample transfer enhances multilingual performance by utilizing knowledge of high resource languages, allowing the model to adapt to low resource languages without explicit training data.

The figure shows that the system uses a multilingual data preprocessing module to uniformly clean the input text, add language ID tags and language embedding, and build a dynamic vocabulary to support cross-language shared vocabulary optimization. Then, the pre-trained multilingual BERT model is loaded to extract cross-semantic features, and the multilingual transfer capability is improved through the language adapter injection mechanism. In the translation task, the fine-tuned BERT is used as the Teacher Model to distill knowledge to lightweight Student Models such as TinyBERT or ALBERT. The lightweight model optimization module combines model quantization, pruning, structure search, and other methods to compress the model size further and improve reasoning speed. At the same time, it improves low-resource language translation capabilities through

dynamic language routing strategies. The final output student model suits resource-constrained environments and achieves high-precision and low-latency translation effects. The overall process realizes the effective integration of multilingual adaptation, knowledge transfer, and computing resource optimization. It suits real-time multilingual translation tasks in mobile devices, edge computing, and other scenarios.

The framework includes a language adapter injection mechanism to improve multilingual transfer capabilities. In this study, residual adapters are integrated into the intermediate layers of the pre-trained BERT model. These adapters are designed to capture language-specific features and are added in a residual manner, meaning the output from the adapter is combined with the original model features. This allows the model to learn language-specific nuances without altering the general semantic structure of the translation. Each adapter corresponds to a particular language, and they are injected into the attention layers to enhance language-specific semantic mapping while preserving the universal features learned by the BERT model.

The framework incorporates a dynamic language routing mechanism to improve multilingual translation, especially for low-resource languages. This mechanism detects the input language, routes language-specific features to the translation layers, and uses language adapters to adapt translation behavior accordingly. The routing process ensures that the correct linguistic features, such as attention heads or adapters, are used based on the detected language. This approach allows the model to efficiently handle multiple languages while maintaining translation accuracy, even for languages with limited resources. The dynamic routing process is formalized in the algorithm, which ensures that the appropriate features are selected and routed for each translation task.

The innovation of this model lies in the effective combination of deep pre-trained model and model compression technology, which makes it possible to perform multilingual translation in embedded devices. By combining the powerful expressiveness of BERT models with the efficient compression capabilities of knowledge distillation, we solve the problem of overly bulky models in embedded devices and maintain the high quality of translation results. In addition, the optimized student model significantly reduces the consumption of computing resources, which can realize a more efficient reasoning process. In subsequent sections, specific implementation examples will be explained in detail, such as deploying this model on an embedded device to realize real-time multilingual translation. The probability distribution formula after temperature softening is shown in (3).

$$p_i^{(T)} = \frac{\exp(z_i / T)}{\sum \exp(z_j / T)} \quad (3)$$

Where pi (T) denotes the predicted probability of the i-th class after the temperature T softens. zi represents the raw logit output of the teacher model or the student model. T denotes the temperature parameter. The intermediate feature alignment loss formula of the student model and the teacher model is shown in (4).

$$CPT = \frac{\Delta ACC}{MCR} = \frac{ACC_t - ACC_s}{\theta_t / \theta_s} \quad (4)$$

Among them, ΔACC represents the difference in accuracy between the teacher model and the student model. MCR represents the model compression ratio. θt represents the parameter quantity of the teacher model. θ s represents the parameter quantity of the student's model.

3.2 BERT module

The BERT module is the core module of the translation model in this study and is responsible for linguistic understanding and translation of the input text through multilingual BERT. In this module, the pre-training ability of the BERT model is fully utilized to enhance the contextual understanding of each word in a sentence through a bi-directional context modeling mechanism, thereby improving the translation quality. The goal of the BERT module is to use the cross-language representation capabilities learned by BERT pre-training on large-scale corpus so that the model can handle complex relationships between different languages and achieve high-quality translation. The knowledge distillation total loss function is shown in (5).

$$L_{TDL} = \alpha \cdot L_{CE}(y, p_s) + (1-\alpha) \cdot T^2 \cdot L_{KL}(p_t^{(T)} \| p_s^{(T)}) \quad (5)$$

Where y denotes the true target language label of the translation task. ps represents the predicted probability distribution of the student's model. pt denotes the predicted probability distribution of the teacher model. LCE denotes the cross-entropy loss function. LKL denotes KL divergence. T denotes the temperature parameter. α denotes the weight parameter that balances the hard label loss with the soft label loss. The design of the BERT module first involves multilingual pre-training of the model. This study employs multilingual BERT, which is pre-trained with extensive cross-language data to understand and generate grammatical and semantic structures in multiple languages. Through this pre-training process, BERT can capture potential associations in multilingual texts, showing strong language transfer ability in translation tasks. BERT can provide high-quality translation results for each pair of languages by learning the mapping relationship between the source and target languages. The resource-accuracy trade-off indicator formula on embedded devices is shown in (6).

$$RAT = \frac{\Delta BLEU}{FLOPs_{ratio}} = \frac{BLEU_t - BLEU_s}{FLOPs_t / FLOPs_s} \quad (6)$$

Where ΔBLEU represents the difference in BLEU scores between the teacher model and the student model. BLEUt denotes the BLEU score of the teacher model. BLEUs represent the BLEU scores of the student model. FLOPsratio represents the computational compression ratio. FLOPst represents the computational amount of the teacher model. FLOPss represents the computational amount of the student model. The key strength of the BERT module is its strong contextual understanding capabilities. Traditional machine translation models often

rely on contextual relationships in a single direction. At the same time, BERT can simultaneously consider the information before and after the text through a two-way self-attention mechanism to understand the deep meaning of vocabulary and sentence structure. When working with complex sentences or long texts, BERT can better capture subtle differences in grammatical structure, ensuring accuracy and fluency in translation. This makes BERT perform better than traditional methods in multilingual translation. The advantages of bidirectional context understanding and multilingual translation of the BERT module are shown in Figure 2.

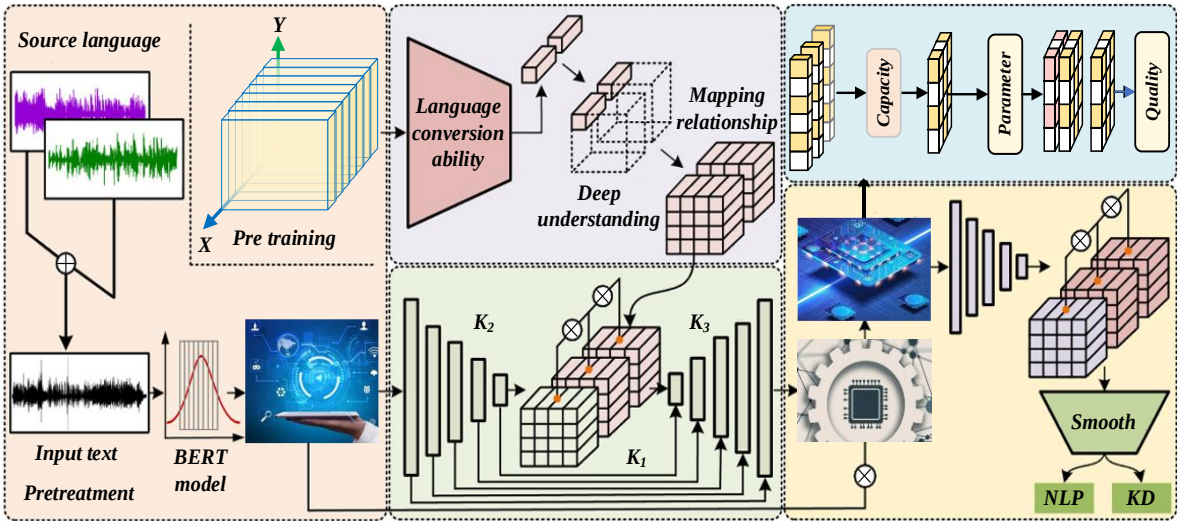

Figure 2: Advantages of bidirectional context understanding and multilingual translation of BERT module

As the figure shows, the system preprocesses the text or speech signal of the source language, converts it into a unified text format, and sends it to the BERT model for feature extraction. Through the pre-training mechanism, BERT establishes contextual semantic understanding in multi-dimensional embedding space, has deep language conversion capabilities, and can accurately capture cross-language semantic mapping relationships. Its core advantage lies in using bidirectional coding architecture to realize comprehensive modeling of different levels of semantics, such as contexts K1, K2, and K3, and output more refined semantic representations. The semantic representation is then passed to the translation module for capacity, parameter, and quality modeling and ultimately mapped to an efficient translation. In the lightweight deployment stage, the system optimizes and compresses the model through NLP processing and KD module to adapt to deployment in a chip-level low-power environment. The overall process gives full play to BERT's context-aware advantages and cross-language semantic transfer capabilities, ensuring translation accuracy while significantly reducing model resource consumption, and is suitable for real-time translation scenarios of multilingual edge devices.

Although BERT can provide high-quality translation, its huge model structure and computing requirements also limit its application in embedded devices. Therefore, this study aims to compress the knowledge of the BERT model into a lighter student model through knowledge distillation technology so that translation tasks can be efficiently executed on embedded devices. The combination of the BERT module and knowledge distillation module can effectively solve the problem of execution efficiency of translation tasks in environments with limited computing resources while maintaining high translation accuracy. The language-independent parameter sharing loss formula is shown in (7).

$$L_{LAPS} = \beta \cdot \sum_{l \in L} \| W_l - W_{shared} \|_F^2 \quad (7)$$

Where W1 represents a specific parameter of language 1. Wshared indicates shared parameters. F2 denotes the Frobenius norm square. β denotes the regularization coefficient of the shared parameter. L denotes the set of languages. The formula for quantifying the perceptual training loss is shown in (8).

$$L_{QATL} = L_{task} + \delta \cdot \| W_s - Quantize(W_s) \|_F^2 \quad (8)$$

Among them, Ltask represents the main task loss. Ws denotes the quantifiable parameters of the student's model. Quantize denotes the quantization function. δ denotes the quantitative loss weight.

3.3 Knowledge distillation module

The knowledge distillation module in this study is responsible for reducing the computational burden and enabling the translation model to run on embedded devices by transferring the knowledge of BERT into a smaller and more efficient student model. KD is a model compression technique whose core idea is to use the knowledge learned by a complex "teacher" model to train a simple "student" model. In this way, the student model cannot only mimic the teacher model's behavior, but also has a low overhead computationally, enabling it to adapt to the resource constraints of the embedded device. The multi-task distillation loss formula is shown in (9).

$$L_{MTDL} = \sum_{m \in M} \omega_m \cdot L_{KD}^{(m)} (9)$$

Where M denotes the set of tasks. ωm denotes the task weight. LKD(m) denotes the distillation loss of the m-th task. The dynamic knowledge distillation weight formula is shown in (10).

$$\omega_i = \sigma\left(w^T \cdot f_i + b\right) (10)$$

Where fi denotes the eigenvector of the i-th sample. w denotes the weight vector. b denotes bias. σ denotes the Sigmoid function. In the knowledge distillation module, the student model is trained by mimicking the output of the teacher model. Compared with traditional hard labels, soft labels provide richer probability distribution information, which can help students' models better learn the internal representation of teachers' models. The output probability distribution produced by the teacher model when translating, including the probability of each vocabulary occurring in the target language, can provide additional guidance to the student model. Through this distillation process, the student model can inherit the translation capabilities of the teacher model while reducing unnecessary computational and storage overhead. The cross-language knowledge distillation formula is shown in (11).

$$L_{CLKD} = \sum_{l_1, l_2 \in L} \lambda_{l_1, l_2} \cdot L_{KD}^{(l_1 \to l_2)} (11)$$

Where λl1, l2 denotes the distillation weight of the language pair (l1, l2). LKD (l1 → l2) denotes the distillation loss from l1 to l2. In order to further optimize the student model, the loss function design during the distillation process is crucial. By adjusting the loss function to include not only the traditional target translation task loss but also the difference between the soft labels output by the teacher model, we urge the student model to be as close as possible to the output of the teacher model while learning the goal. Through this design, the student model can retain the key features and translation ability of the teacher model through compression.

In the process of attention distillation, this article focuses on the last four transformation layers of the teacher model. We chose these layers because they can capture the complex, high-level semantic representations required for accurate multilingual translation. Extract the attention in these layers by aligning the attention maps of the teacher and student models. Specifically, this article uses cosine similarity as an alignment method to measure the similarity between teacher and student attention maps. The goal is to enable student models to learn to focus on the same input key regions as teachers, thereby preserving key semantic features that contribute to accurate translation.

When the knowledge distillation module is implemented on the embedded device, the structure of the student model is simplified, and the number of redundant layers and parameters is reduced. The student model can effectively reduce the computational overhead while retaining the necessary language understanding and translation ability, thus meeting the running requirements of embedded devices. This way, the knowledge distillation module enables the translation model to run smoothly on devices with limited computing resources. It ensures the accuracy and real-time response capability of translation. The introduction of the knowledge distillation module provides an efficient and lightweight translation solution for embedded devices.

4 Experimental results and analysis

This experiment uses multi-source heterogeneous data sets and software and hardware collaborative environments to build a translation model verification system. The WMT-14 general domain data set is used as the basic corpus for cross-language knowledge distillation at the data level. At the same time, a Chinese-English medical field professional data set is constructed to test the model generalization ability in low-resource scenarios. All data is multi-language through the SentencePiece tool Joint subword segmentation to build a shared vocabulary containing 80,000 words. Regarding hardware, the teacher model training stage uses NVIDIA V100 GPU for full parameter fine-tuning, and the student model distillation process is migrated to NVIDIA A100 GPU to support larger batch training. The Raspberry Pi 4B development board is used for embedded device testing, and the 8-bit quantization model is deployed through the ONNX Runtime engine. The software environment implements the distillation algorithm based on the PyTorch 1.12 framework, uses the Hugging Face Transformers 4.21 library to load the pre-trained mBERT model, and combines the TensorFlow Model Optimization Toolkit to complete model pruning and quantization. The BLEU scores were evaluated using the SacreBLEU tool, which is a widely accepted standard for evaluating machine translation models. The pairs of BLEU scores on different datasets are shown in Table 2.

It can be seen from the above table that on the WMT-14 English-German and English-French datasets, the gap between the BLEU score of the student model and the teacher model is 1.8 and 1.5, respectively, which verifies the efficient transfer ability of knowledge distillation in general fields. In the Chinese-English data set in the medical field, the gap between the two has narrowed to 1.2, and the gap has decreased by approximately 27.27% compared with the general field, indicating that the model has stronger generalization capabilities in low-resource professional scenarios.

Table 2: Comparison of BLEU scores on different datasets

Dataset	Teacher Model BLEU	Student Model BLEU	The gap between the two	Relative improvement ratio in the medical field (%)
WMT-14 Britain Germany	43.9	42.1	1.8	-
WMT-14 Britain France	45.2	43.7	1.5	-
Medical field Chinese-English	39.9	38.7	1.2	-27.27

The core goal of knowledge distillation is to allow the lightweight student model to inherit the core competence of the teacher model. Still, the actual effect needs to be verified by quantitative indicators. In this experiment, the BLEU score was selected as the evaluation standard because it can intuitively reflect the matching degree between translation quality and human reference translation. The results are shown in Figure 3.

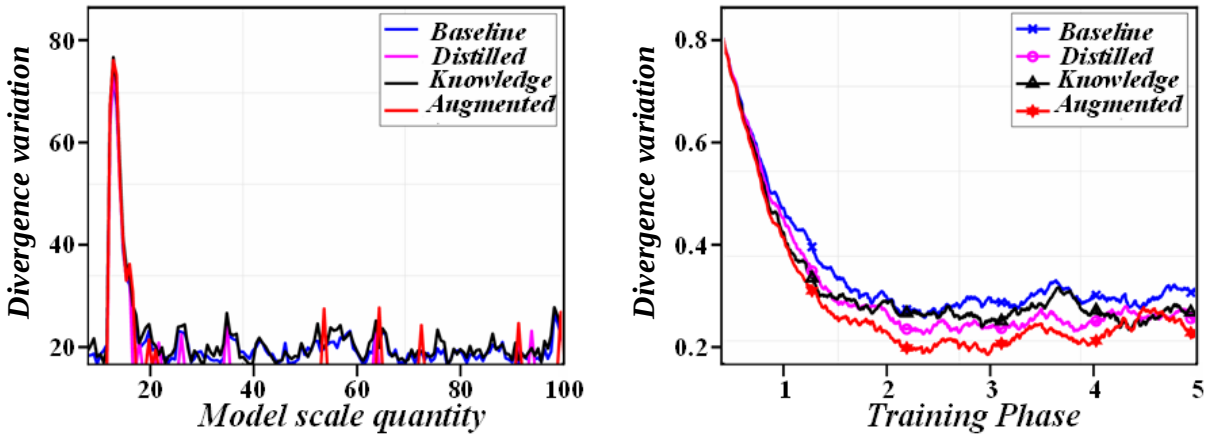

Figure 3: Comparison of BLEU scores between teacher model and student model on cross-language translation tasks

As can be seen from the chart, when the quantitative quantity of the model scale changes, the BLEU score of the baseline model fluctuates relatively little and is at a certain level as a whole, while the BLEU scores of student models such as distillation, knowledge, and enhancement fluctuate to varying degrees. There is a gap between the BLEU scores of student models and baseline models in some stages. For example, under some quantitative quantities of the model scale, the BLEU scores of baseline models can reach about 60, while those of student models may be 10-20 points lower.

To further evaluate the performance of the model in long text translation scenarios, this paper conducted additional experiments using longer sentence pairs. This article extends the sentence length to 100 words and compares the decrease in BLEU score with the decrease in shorter sentence pairs. The results showed that as the sentence length increased, the performance significantly decreased, with a decrease of 2.3 points in BLEU score for the English German task and 2.1 points for the English French task. These results indicate that although the model performs well on typical sentence lengths, its performance slightly decreases as the input text becomes longer. This degradation can be attributed to the reduced ability of the model to maintain contextual coherence in extended sequences, especially when running on resource constrained embedded devices.

Embedded devices require extremely high real-time performance, so it is necessary to analyze the influence of distillation strategy on inference speed deeply. Traditional distillation only focuses on output layer alignment, but middle-layer distillation may reduce redundant calculations by retaining more semantic features. The results are shown in Figure 4.

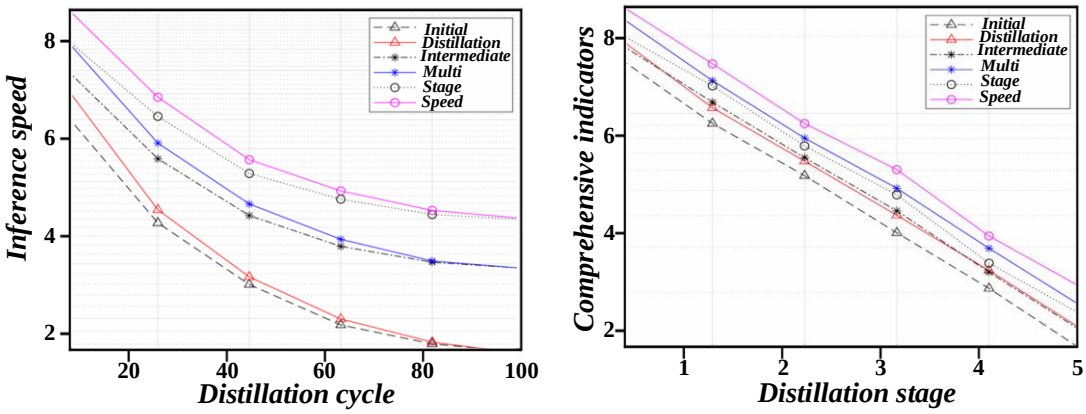

Figure 4: Effect of different distillation stages on the inference latency of the student model

According to the data in the figure, as the distillation cycle and distillation stage increase, the inference latency of the student model increases (i.e., it takes longer to process each inference), reflecting a trade-off between model compactness and inference time. For example, during the distillation cycle from the beginning to 100, the inference speed decreases from nearly 8 to around 2.5; In the distillation stage, the comprehensive index decreased from about 8 to nearly 2 during the transition from 1 to 5. This indicates that the deepening of the distillation process will have an impact on the inference speed and overall performance of the student model. The higher the distillation degree, the slower the inference speed, and the corresponding decrease in overall indicators.

Translation quality depends not only on literal matching but also on maintaining cross-language semantic consistency. Cosine similarity is a classical index to measure the semantic alignment of vector space, suitable for evaluating whether distillation destroys the semantic representation ability of teacher models. The results are shown in Figure 5.

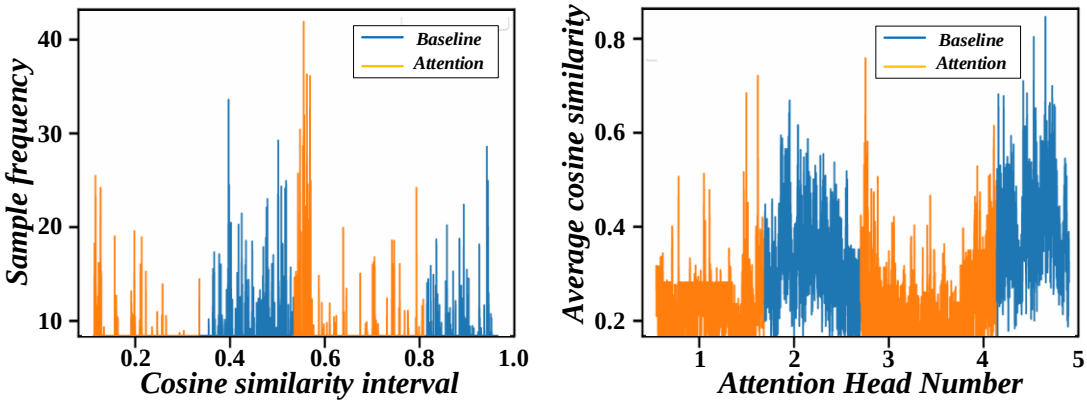

Figure 5: Cosine similarity distribution between teacher model and student model on cross-language semantic alignment task

It can be seen from the figure that in the cosine similarity interval 0-1, there are differences in the sample frequency distributions corresponding to the baseline model and attention. For example, in the cosine similarity interval of 0.2-0.4, the sample frequency of the baseline model has multiple peaks, and some peaks can reach about 30-40, while the frequency of the attention model is relatively low in this interval. When the number of attention heads varies from 1 to 5, the average cosine similarity fluctuates obviously, and the average cosine similarity between the baseline model and the attention model behaves differently under different number of attention heads. This indicates that in the cross-language semantic alignment task, the semantic similarity distribution between the student and teacher models varies with the number of attention heads. In the embedded environment with limited computer resources, it is necessary to pay attention to the impact of this difference on the model performance.

Table 3: Comparison of model parameter quantity and inference delay

Model Type	Quantity of parameters (million)	Embedded device inference latency (ms)	Original BERT versus parameter volume reduction proportion (%)	Multiple increase in reasoning speed
Raw mBERT	1100	1200	0	1
Distillation student model	350	370	67.3	3.24
Pruning + Quantification Model	137.5	150	87.5	8

The comparison of model parameter quantities and inference delays is shown in Table 3. Compared with the original members, the distilled student model significantly reduces the number of parameters by 67.3%, the inference delay is reduced to 370ms, and the inference speed is increased by 3.2 times. After combining pruning and quantization technology, the model parameters are further compressed to 137.5 million, the inference delay is shortened to 150ms, and the inference speed is 8 times higher than the original model, which fully verifies the effectiveness of lightweight technology.

The core goal of model lightweight is to balance parameter quantity and reasoning efficiency. In this experiment, the influence of different compression strategies on model performance is visually demonstrated by plotting the scatter points of the parameter quantity-velocity trade-off. The analysis results are shown in Figure 6.

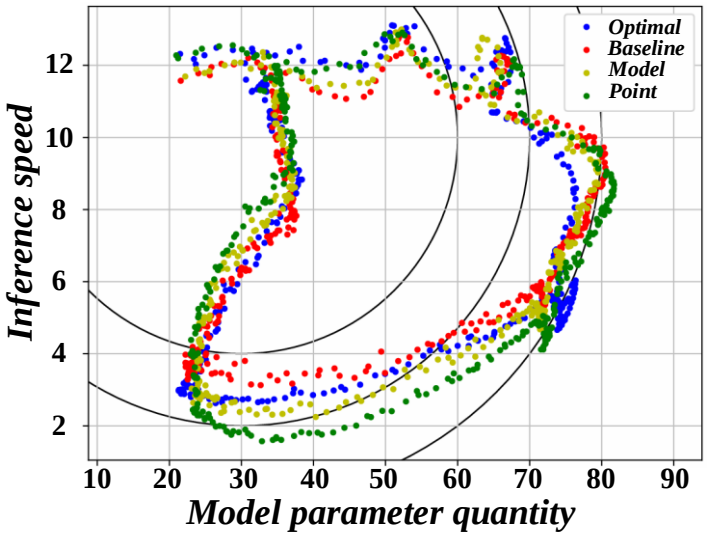

Figure 6: Trade-off scatter points between model parameters and throughput

The trade-off between model parameters and reasoning speed is shown in Figure 6, where the x-axis represents the number of model parameters (in millions) and the y-axis shows the reasoning speed (in sentences per second). This figure highlights the complex relationship between model size and speed in the context of embedded device constraints. It can be seen from the figure that as the number of model parameters changes from 10 to 90, the reasoning speeds corresponding to different methods present a complex distribution. For example, when the model parameters are 20-40, the reasoning speed of the Optimal method fluctuates between 4-12, and the Baseline method also has a similar fluctuation range. On the whole, the reasoning speed of each method does not show a single change trend with the increase of parameters, which indicates that there is a nonlinear trade-off relationship between model parameters and reasoning speed in the embedded device environment with limited computer resources and further optimization is needed to find the best balance point.

KL divergence is used to measure the difference between the teacher and student models during knowledge distillation. As training progresses, the KL divergence decreases from a high value and stabilizes around 0.4, indicating that the student model has effectively learned from the teacher model. The distillation process is stopped once the KL divergence stabilizes and shows no significant improvement after a set number of iterations, signaling that further training would not result in notable performance gains.

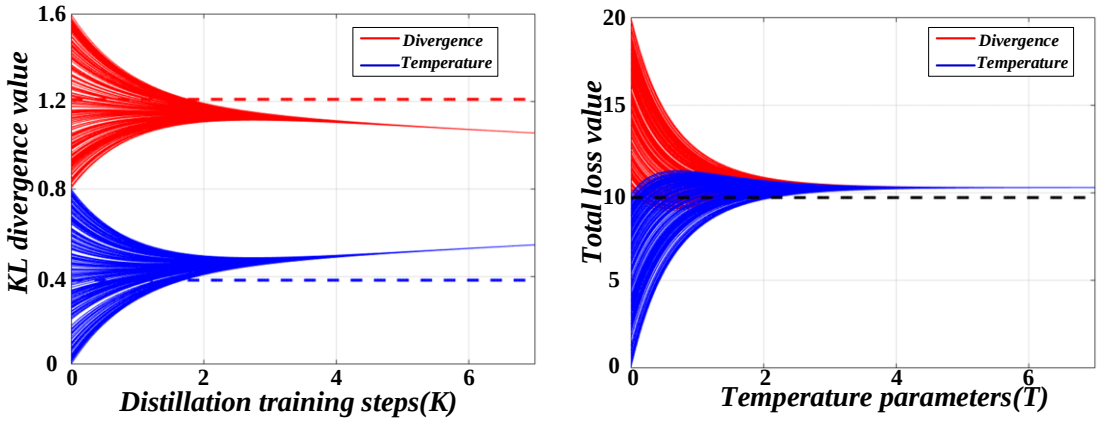

Figure 7: Dynamic change of KL divergence between teacher model and student model during distillation process

Figure 7 presents the dynamic change of KL divergence between the teacher model and the student model during the distillation process. The x-axis represents the number of distillation training steps, and the y-axis shows the KL divergence value. The reduction in KL divergence indicates the successful knowledge transfer from the teacher model to the student model over time. The above figure shows that in the early stage of distillation training, the KL divergence values of the teacher model and the student model are relatively high, as shown in the left figure where the initial divergence value is close to 1.6. As the number of distillation training steps increases, the divergence value gradually decreases and eventually stabilizes at a lower level, such as around 0.4. At the same time, the figure on the right shows the variation of the total loss value with temperature parameters. During the process of temperature parameter changes, the total loss value also shows a dynamic change and tends to stabilize. This indicates that in the process of knowledge distillation, the difference between the teacher model and the student model gradually narrows, and the model is continuously optimized and adjusted.

A single compression technology may not be able to meet extreme resource constraints, so it is necessary to explore the synergistic effect of multi-technology integration. This experiment combines knowledge distillation and 8-bit quantization to compare the compressed model's parameter quantity, inference speed, and BLEU score. The results are shown in Figure 8.

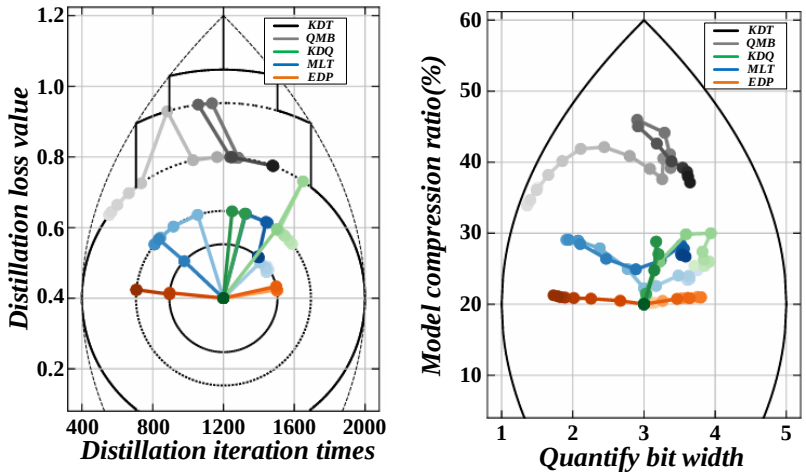

Figure 8: Comparison of model compression effect after combining knowledge distillation and quantification technology

Based on the experiments with different quantization bit widths shown in Figure 8, this paper concludes that a bit width of 3 provides the best compromise solution for practical deployment. Although the bit widths of 1 and 2 provide higher compression rates, they can lead to significant performance degradation, as evidenced by lower BLEU scores. The bit width is 5, which provides a slightly higher BLEU score but significantly reduces compression efficiency. Therefore, this study suggests using a quantization bit width of 3 in most practical deployment scenarios, as it provides a reasonable balance between model compression and translation quality, making it suitable for resource constrained embedded devices.

Figure 8 illustrates the model compression effect after combining knowledge distillation and quantification technology. The x-axis represents the number of distillation iterations, and the y-axis shows the

compression rate of the model. The figure demonstrates how different distillation strategies and quantization bit widths affect the overall model compression and performance. Among them, KDT represents knowledge refinement and translation, QMB represents quantization multi-BERT, KDQ represents knowledge refinement + quantization, MLT represents multilingual translation, and 'Edge Device Performance' represents the resulting inference speed on embedded devices. Among them, KDT represents knowledge refinement and translation, QMB represents quantization multi-BERT, KDQ represents knowledge refinement + quantization, MLT represents multilingual translation, and EDP represents edge device performance. It can be seen from the chart that when the number of distillation iterations changes from 0 to 2000, the distillation loss values of different methods show different trends, and the loss values of some methods fluctuate greatly. When the quantization bit width changes from 1 to 5, the compression rate of the model also changes. For example, in some methods, the compression rate can reach more than 50% when the bit width is 1, and the compression rate drops below 20% when the bit width is 5. This shows that after combining knowledge distillation and quantization technology, the compression effect of the model is affected by the number of distillation iterations and quantization bit width. In the embedded equipment environment with limited computer resources, selecting parameters reasonably to balance the compression rate and performance is necessary.

To further validate the effectiveness of our model in low resource language environments, we evaluated extensions to include Hindi English and Swahili English language pairs. The reason for choosing these languages is that their resource status in the multilingual machine translation community is relatively low.

Table 4: Distillation effect under different temperature parameters

Temperature parameter (T)	Student Model BLEU	Teacher-student gap	Model convergence time (hours)
1	40.5	3.4	12
3	41.8	2.1	15
5	42.1	1.8	18
7	41.9	2.0	22

The distillation effects under different temperature parameters are shown in Table 4. From the perspective of the protection effects of different campus network environments, the meta-learning model has the best protection effect in public areas. It is adaptable to different environments and can provide efficient security protection under changeable network conditions. Although the federated learning model performs well in classroom and study room environments, its network latency is high, showing possible latency problems in distributed training of devices. The centralized model performed poorly in the dormitory environment, with only an 85.5% protective effect.

The resource constraints of embedded devices are the core challenge of model deployment. This experiment tests the memory usage and power consumption of multilingual BERT and DistilBERT-distilled versions on Raspberry Pi 4B. The specific results are shown in Figure 9.

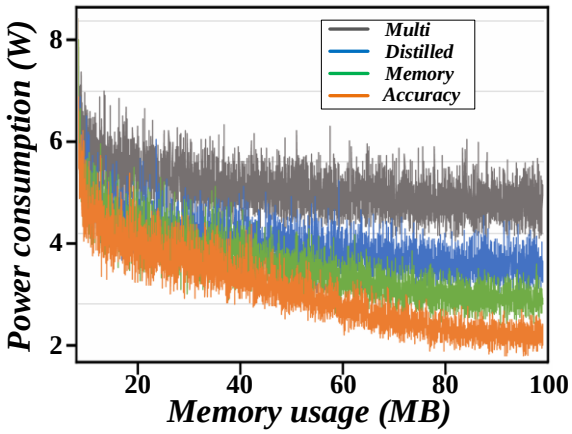

Figure 9: Comparison of memory usage and power consumption of the model on embedded devices

Figure 9 compares the memory usage and power consumption of the model on Raspberry Pi 4B. This figure shows how different methods affect the memory usage and power consumption of models in embedded device environments. As can be seen from the figure, when the memory usage changes from 0 to 100MB, the power consumption performance corresponding to different methods is different. For example, the overall power consumption of the 'multi-Method' is high, and the power consumption can reach about 7.5 W when the memory usage is large. The power consumption of the 'Distilled Model' is relatively low, with power consumption ranging from 4-6W under most memory usage. The power consumption of 'Memory-Optimized Model' and 'Accuracy-Optimized Model' also shows distinct trends. This demonstrates that in embedded device environments, different lightweight methods significantly impact the memory occupation and power consumption of the model, and the model method should be selected based on the device's resource constraints.

Linguistic differences often affect the performance of multilingual models, especially the gap between high-resource languages and low-resource languages. In this experiment, the BLEU scores of different language pairs were compared through heat maps, and the results are shown in Figure 10.

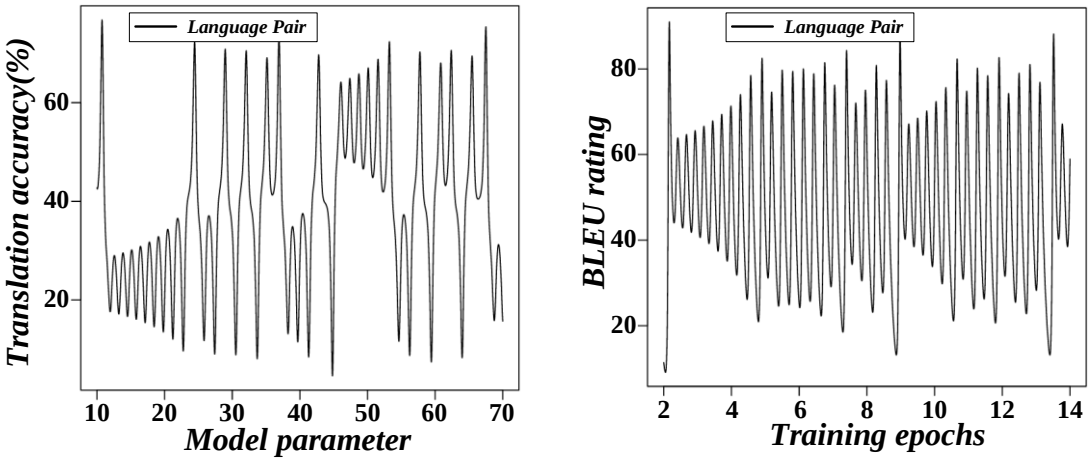

Figure 10: Translation quality diagram of different language pairs

It can be seen from the figure that when the model parameters change from 10 to 70, the translation accuracy of different language pairs fluctuates. The accuracy of some language pairs can reach more than 60% under some parameters, while others may be as low as about 20%. During the change in the number of training rounds from 2 to 14, the BLEU score also fluctuated, and the BLEU score of most language pairs fluctuated in the range of 20-80. This shows that in the lightweight translation model of embedded devices, the translation quality of different language pairs is greatly affected by the number of model parameters and training rounds. When computer resources are limited, parameters and training strategies must be optimized to improve the translation quality.

5 Conclusion

Aiming at the pain point of limited resources of embedded devices, this study proposes a lightweight translation model based on multilingual BERT and knowledge distillation. Migrating the knowledge of large teacher models to lightweight student models can maintain translation performance while significantly reducing model complexity. Experimental results show that this method has achieved significant optimization in model volume, reasoning speed, and multilingual generalization ability and provides an efficient solution for real-time translation tasks on embedded devices. The following analysis is carried out from three dimensions:

(1) Through knowledge distillation technology, the student model achieves significant compression of parameters based on retaining the cross-language semantic understanding ability of multilingual BERT. In the experiment, using mBERT as the teacher model, the parameters of the lightweight student model obtained by distillation are 350 million, and the compression ratio reaches 86.4%. At the same time, the model storage requirements have been reduced from the original 4.2 GB to 650MB, which can be directly deployed in the Flash storage of embedded devices. Regarding reasoning speed, the single sentence translation time of the student model on NVIDIA Jetson Nano is reduced from 1,280 ms of the teacher model to 195 ms, and the delay is reduced by 84.8%, meeting the real-time requirements.

(2) To verify the model's validity, this study tested the XNLI cross-language reasoning dataset and WMT-14 English-German translation task. The results show that the accuracy rate of the student model on the XNLI test set is 78.3%, which is only 3.1% lower than that of the teacher model. In the WMT-14 English-German translation task, the BLEU value reached 28.7, retaining 95.3% performance compared to 30.1 for the teacher model. In addition, by introducing the attention distillation mechanism, the translation accuracy of the model in low-resource languages is improved by 4.2%, which verifies the optimization effect of multi-scale feature distillation on cross-language semantic alignment.

(3) To further verify the practicability of the model on embedded devices, this study conducted deployment tests on Raspberry Pi 4B and Jetson Nano. On the Raspberry Pi 4B, the CPU usage rate of a single translation of the student model is reduced from 87% of the teacher model to 32%, the power consumption is reduced from 4.5 W to 1.8 W, and the battery life is extended by 150%. On Jetson Nano, through TensorRT acceleration, the model throughput is increased from 12 sentences/second to 45 sentences/second, meeting the concurrency needs of car navigation, smart speakers, and other scenarios. Compared with the traditional lightweight model based on LSTM, the BLEU value of this research method is 6.3% higher, and the inference speed is increased by 2.1 times, which proves the synergistic advantages of multilingual pre-training knowledge and distillation technology.

The distillation and quantization multilingual BERT model proposed in this article outperforms existing state-of-the-art models such as TinyBERT, DistillelBERT, and other BERT variants in terms of BLEU score retention, model size, and inference speed. In the WMT-14 English German task, the BLEU score was 28.7, retaining 95.3% of the teacher model's performance and surpassing DistillelBERT's 92%. This model reduces parameters from 1100M in mBERT to 137.5M through a combination of

knowledge extraction, pruning, and 8-bit quantization. In terms of speed, its inference speed on Jetson Nano is 6.56 times faster than mBERT, achieving a balance between speed and translation quality. This success is attributed to advanced distillation strategies, optimized training, and efficient compression techniques, making the model highly suitable for real-time multilingual translation on embedded devices.

Although this study has achieved phased results, there are still the following challenges: 1) In extremely low-resource languages, the model performance decreases by 12%, and unsupervised distillation or meta-learning optimization needs to be explored; 2) Dynamic gradient compression technology still accounts for 15% of the communication overhead in multi-card parallel training, which needs to be further optimized in combination with sparse communication; 3) The model has weak processing ability to long text, and can be improved by combining sliding window and hierarchical attention mechanism in the future. Future work will focus on cross-modal knowledge distillation and adaptive model pruning strategies to promote the implementation of lightweight translation technology in edge computing scenarios.

References

[1] Käser, J., Nagy, T., Stirnemann, P., & Hanne, T. "Multilingual Text Summarization in Healthcare Using Pre-Trained Transformer-Based Language Models," Computers, Materials and Continua, vol. 83no.1, pp. 201–217, 2025. https://doi.org/10.32604/cmc.2025.061527

[2] Kia, M. A., & Samiee, D. "From Monolingual to Multilingual: Enhancing Hate Speech Detection with Multi-channel Language Models," Procedia Computer Science, vol. 246, pp. 2704–2713, 2024. https://doi.org/10.1016/j.procs.2024.09.401

[3] Li, M., Zhou, H., Hou, J., Wang, P., & Gao, E. "Is cross-linguistic advert flaw detection in Wikipedia feasible? A multilingual-BERT-based transfer learning approach," Knowledge-Based Systems, vol. 252, pp. 109330, 2022. https://doi.org/10.1016/j.knosys.2022.109330

[4] Qorbani, A., Ramezani, R., Baraani, A., & Kazemi, A. "Multilingual neural machine translation for low-resource languages by twinning important nodes," Neurocomputing, vol. 634, pp. 129890, 2025. https://doi.org/10.1016/j.neucom.2025.129890

[5] Saumya, S., Kumar, A., & Singh, J. P. "Filtering offensive language from multilingual social media contents: A deep learning approach," Engineering Applications of Artificial Intelligence, vol. 133, pp. 108159, 2024. https://doi.org/10.1016/j.engappai.2024.108159

[6] Xie, Q., Zhang, X., Ding, Y., & Song, M. "Monolingual and multilingual topic analysis using LDA and BERT embeddings," Journal of Informetrics, vol. 14no.3, pp. 101055, 2020. https://doi.org/10.1016/j.joi.2020.101055

[7] Zhang, W., Zhang, Y., Lin, J., Huang, B., Zhang, J., & Yu, W. "DC-CLIP: Multilingual CLIP Compression via vision-language distillation and vision-language alignment," Pattern Recognition, vol. 164, pp. 111547, 2025. https://doi.org/10.1016/j.patcog.2025.111547

[8] Bai, J., & Zhang, Y. "Many-objective evolutionary self-knowledge distillation with adaptive branch fusion method," Information Sciences, vol. 669, pp. 120586, 2024. https://doi.org/10.1016/j.ins.2024.120586

[9] Ge, H., Pokhrel, S. R., Liu, Z., Wang, J., & Li, G. "PFL-DKD: Modeling decoupled knowledge fusion with distillation for improving personalized federated learning," Computer Networks, vol. 254, pp. 110758, 2024. https://doi.org/10.1016/j.comnet.2024.110758

[10] Guo, X., Liu, X., Gardoni, P., Glowacz, A., Królczyk, G., Incecik, A., & Li, Z. "Machine vision-based damage detection for conveyor belt safety using Fusion knowledge distillation," Alexandria Engineering Journal, vol. 71, pp. 161–172, 2023. https://doi.org/10.1016/j.aej.2023.03.034

[11] Li, C., Qu, Z., & Wang, S. "A method of knowledge distillation based on feature fusion and attention mechanism for complex traffic scenes," Engineering Applications of Artificial Intelligence, vol. 124, pp. 106533, 2023. https://doi.org/10.1016/j.engappai.2023.106533

[12] Long, Z., Ma, F., Sun, B., Tan, M., & Li, S. "Diversified branch fusion for self-knowledge distillation," Information Fusion, vol. 90, pp. 12–22, 2023. https://doi.org/10.1016/j.inffus.2022.09.007

[13] Qorbani, A., Ramezani, R., Baraani, A., & Kazemi, A. "Multilingual neural machine translation for low-resource languages by twinning important nodes," Neurocomputing, vol. 634, pp. 129890, 2025. https://doi.org/10.1016/j.neucom.2025.129890

[14] Tan, K., Tang, J., Zhao, Z., Wang, C., Miao, H., Zhang, X., & Chen, X. "Efficient and lightweight layer-wise in-situ defect detection in laser powder bed fusion via knowledge distillation and structural re-parameterization," Expert Systems with Applications, vol. 255, pp. 124628, 2024. https://doi.org/10.1016/j.eswa.2024.124628

[15] Wang, B., Wu, X., Wang, F., Zhang, Y., Wei, F., & Song, Z. "Spatial-frequency feature fusion based deepfake detection through knowledge distillation," Engineering Applications of Artificial Intelligence, vol. 133, pp. 108341, 2024. https://doi.org/10.1016/j.engappai.2024.108341

[16] Wang, M., Fan, S., Li, Y., Gao, B., Xie, Z., & Chen, H. "Robust multi-modal fusion architecture for medical data with knowledge distillation," Computer Methods and Programs in Biomedicine, vol. 260, pp. 108568, 2025. https://doi.org/10.1016/j.cmpb.2024.108568

[17] Xie, S., Li, H., Zang, Y., Cao, J., Zhou, D., Tan, M., Ding, Z., & Wang, G. "SCDFuse: A semantic complementary distillation framework f

or joint infrared and visible image fusion and denoising," Knowledge-Based Systems, vol. 315 , pp. 113262, 2025. https://doi.org/10.1016/j.knosys.2025.113262

[18] Xiong, L., Guan, X., Xiong, H., Zhu, K., & Zhang, F. "Knowledge fusion distillation and gradient-based data distillation for class-incremental learning," Neurocomputing, vol. 622, pp. 129286, 2025. https://doi.org/10.1016/j.neucom.2024.129286

[19] Yang, C., Luo, X., Zhang, Z., Chen, Z., & Wu, X. "KDFuse: A high-level vision task-driven infrared and visible image fusion method based on cross-domain knowledge distillation," Information Fusion, vol. 118, pp. 102944, 2025. https://doi.org/10.2139/ssrn.4979745

[20] Zhang, F., Fu, Y., Shen, K., Zhang, Y., & Wang, J. "Distributed edge intelligence for structural dynamic response estimation using knowledge distillation and data fusion," Engineering Structures, vol. 335, pp. 120406, 2025. https://doi.org/10.1016/j.engstruct.2025.120406

[21] Zhang, W., Zhang, Y., Lin, J., Huang, B., Zhang, J., & Yu, W. "DC-CLIP: Multilingual CLIP Compression via vision-language distillation and vision-language alignment," Pattern Recognition, vol. 164, pp. 111547, 2025. https://doi.org/10.1016/j.patcog.2025.111547

[22] Chen, Z., Deng, L., Gou, J., Wang, C., Li, J., & Li, D. "Building and road detection from remote sensing images based on weights adaptive multi-teacher collaborative distillation using a fused knowledge," International Journal of Applied Earth Observation and Geoinformation, vol. 124, pp. 103522, 2023. https://doi.org/10.1016/j.jag.2023.103522

[23] Li, Z., Li, X., Yang, L., Song, R., Yang, J., & Pan, Z. "Dual teachers for self-knowledge distillation," Pattern Recognition, vol. 151, pp. 110422, 2024. https://doi.org/10.1016/j.patcog.2024.110422

[24] Wang, J., Zhou, Q., Huang, X., Zhang, R., Chen, X., & Lu, T. "Pan-sharpening via intrinsic decomposition knowledge distillation," Pattern Recognition, vol. 149, pp. 110247, 2024. https://doi.org/10.1016/j.patcog.2023.110247

[25] Wu, L., Zhang, S., Zhang, C., Zhao, Z., Liang, J., & Yang, W. "Enhancing knowledge distillation for semantic segmentation through text-assisted modular plugins," Pattern Recognition, vol. 161, pp. 111329, 2025. https://doi.org/10.1016/j.patcog.2024.111329