Received 00 xxxx 0000, accepted 00 xxxx 0000, date of publication 00 xxxx 0000, date of current version 00 xxxx 0000

Digital Object Identifier 00.0000/xxxx.0000.Doi Number

Application of lightweight Transformer in real-time text translation on mobile devices

Xin Li[1], Mingyu Zhou[2]

[1]Shandong Normal University, Jinan 250358, PR China

[2]School of Communication and Electronic Engineering, Shandong Normal University, Jinan 250358, PR China

Corresponding author: Xin Li (e-mail: lixin926@ 163.com).

ABSTRACT Real-time text translation on mobile devices is a powerful support for cross-language communication. However, traditional Transformer models suffer from adaptation problems such as numerous parameters, high computational cost, and long inference latency. Lightweight Transformers, through simplified structure, quantization pruning, and improved inference, reduce resource consumption while maintaining translation accuracy and providing an efficient solution for mobile devices. This paper systematically reviews the research status and application value of lightweight Transformers in real-time text translation on mobile devices, and carefully explores relevant model optimization and deployment adaptation techniques, hoping to provide some reference for related technology development and use, thereby promoting more convenient and widespread cross-language communication.

INDEX TERMS Lightweight, Transformer mobile application, Real-time text translation.

I. INTRODUCTION

In the wave of globalization, real-time text translation on mobile devices has become an essential tool for people's travel, work, and social interactions. The core requirement is to achieve translation results with "low latency, high accuracy, and low power consumption" within limited device resources. Traditional Transformer models achieve ideal translation results through self-attention mechanisms, but due to the hundreds of millions of parameters and complex computational process, they are difficult to adapt to the storage, computing power, and power consumption requirements of mobile devices. Lightweight Transformer technology resolves the contradiction between high performance and lightweight design, optimizing various aspects to enable real-time text translation on mobile devices such as smartphones. This paper studies and analyzes the application value and implementation path of lightweight Transformer in real-time text translation on mobile devices, elaborating on it from three aspects: current status, significance, and strategies.

II. CURRENT STATUS OF RESEARCH AND APPLICATION OF LIGHTWEIGHT TRANSFORMER IN REAL-TIME TEXT TRANSLATION ON MOBILE DEVICES

A. CURRENT STATUS OF MODEL COMPRESSION TECHNOLOGY RESEARCH

Currently, the compression of lightweight Transformer models mainly focuses on parameter simplification, with mainstream approaches including weight sharing, low-rank decomposition, and knowledge distillation. Weight sharing involves sharing parameters between different layers or heads to reduce redundant storage; for example, ALBERT splits the embedding layer parameters into small matrices, reducing the number of parameters by more than 70%. Low-rank decomposition transforms high-dimensional weights into low-dimensional matrix products, compressing the size with minimal accuracy loss. Knowledge distillation utilizes a large model as a teacher model to guide a lightweight student model in learning feature representations. A balance between accuracy and efficiency has been achieved in the lightweighting of models like Transformer-XL and T5, but distillation strategies for mobile scenarios still need to be adjusted based on real-time requirements.

B. CURRENT STATUS OF MOBILE DEPLOYMENT OPTIMIZATION RESEARCH

Mobile deployment optimization aims to improve inference efficiency, mainly focusing on framework adaptation, operator optimization, and hardware acceleration. Inference frameworks such as TensorFlowLite and PyTorchMobile already support edge deployment of Transformer models, but

core operators such as self-attention still require optimization. Operator optimization combines multiple matrix multiplications in multi-head attention into a single computation, simplifying mathematical operations and reducing latency. Hardware acceleration utilizes heterogeneous computing units such as mobile GPUs and NPUs; some flagship models can already achieve lightweight Transformer parallel inference, but hardware compatibility still varies among mid-to-low-end devices.

C. CURRENT STATUS OF REAL-TIME PERFORMANCE IMPROVEMENT RESEARCH

Research on real-time performance mainly focuses on inference latency and response speed. Techniques include sequence length control, incremental inference, and pre-computation. Sequence length control dynamically truncates long texts or translates in segments, reducing processing time per sentence, but ensuring truncation accuracy and translation integrity. Incremental inference only performs calculations on newly added parts, reusing previously calculated results, making it suitable for conversational translation. Pre-computation caches unchanging computational loads in the model beforehand, directly calling them during inference, reducing real-time computational overhead; however, the storage space occupied by the pre-computation cache must be strictly limited to adapt to mobile devices.

D. CURRENT STATUS OF PRACTICAL APPLICATION SCENARIOS

Lightweight Transformer has been used in mainstream mobile translation applications, such as Baidu Translate and Youdao Translate, which use lightweight models for offline real-time translation. Application scenarios include text scanning translation, dialogue translation, cross-border e-commerce product translation, etc. When scanning, it will also combine OCR technology to use lightweight models to achieve the effect of instant translation by taking a picture. However, there are still some shortcomings. For low-end and mid-range devices, the reasoning speed needs to be improved. The translation of minor languages needs to solve the problem of lightweighting, and the fluency of complex sentence translation also needs to be improved [1].

III. THE SIGNIFICANCE OF APPLYING LIGHTWEIGHT TRANSFORMERS TO REAL-TIME TEXT TRANSLATION ON MOBILE DEVICES

A. ENHANCING THE USER EXPERIENCE ON MOBILE DEVICES

The lightweight Transformer significantly reduces the device resources required for model runtime, resolving the issues of slow startup, lag, and high power consumption of traditional models on mobile devices. Users can perform local real-time translation without waiting for model loading and cloud response. Offline translation technology enhances application convenience in situations without network or with poor network signal. Low power consumption results in less heat generation and less power consumption, eliminating concerns about prolonged use of the translation function and improving the user experience [2].

B. PROMOTING THE IMPLEMENTATION OF NATURAL LANGUAGE PROCESSING TECHNOLOGY

The lightweight Transformer opens the door to large-scale application of NLP technology on mobile devices. Traditional heavy-duty Transformer models are limited by hardware resources and can only be deployed in the cloud, while the lightweight model achieves a "device-side" breakthrough, enabling NLP technology to move from the cloud to the terminal. This not only enriches the application scenarios of real-time text translation but also provides a reference for the mobile implementation of other NLP tasks such as speech recognition and sentiment analysis, accelerating the pace of NLP technology moving from laboratory research to practical application and allowing artificial intelligence technology to benefit the public.

C. EMPOWERING CROSS-LANGUAGE COMMUNICATION IN MULTIPLE INDUSTRIES

In industries such as cross-border tourism, international trade, and foreign education, the mobile real-time text translation brought by the lightweight Transformer has become a medium for cross-language communication. When traveling, tourists can use their mobile phones to translate road signs and menus to eliminate language barriers; business people in foreign trade can also use their mobile phones to translate conversations in real time to improve work efficiency; and when studying in life, they can also use their mobile phones to translate in real time to help themselves learn better. This technology enables cross-language communication between various industries, reduces communication costs, and is conducive to globalization[3].

IV. OPTIMIZATION STRATEGIES FOR LIGHTWEIGHT TRANSFORMER IN REAL-TIME TEXT TRANSLATION ON MOBILE DEVICES

A. MODEL SIMPLIFICATION STRATEGIES

Model simplification primarily reduces resource consumption through three aspects: redundancy removal, computational simplification, and balancing accuracy with lightweight design. Replacing fully connected layers in the transformer encoder and decoder with depthwise separable convolutions, and decomposing standard convolutions into depthwise and pointwise convolutions, reduces computation and parameter count by approximately 8-9 times (MobileBERT was inspired by this idea, compressing the original BERT model parameters to 14MB with a 2.6% decrease in accuracy). Optimization using self-attention mechanisms, such as sparse attention replacing full attention, sliding window attention, and local + global attention, can all reduce computation in long text translation. Using sliding

window attention (such as Longformer) and local + global attention (BigBird) to reduce computation at important positions in the text can reduce computation by more than 60% in long text translation. Reasonably reducing... The number of encoder/decoder layers and the number of heads in multi-head attention were compared experimentally to find the best structure suitable for mobile devices by comparing the translation accuracy and computational cost under different combinations of layers (2-6 layers) and heads (2-8 heads). For example, a simplified structure of "3-layer encoder + 3-layer decoder + 4-head attention" was adopted for low-end devices, which significantly reduced resource consumption while ensuring acceptable accuracy. A bottleneck layer was added to compress features, and a low-dimensional feature mapping layer was added between the encoder and decoder to reduce the computational cost when transferring features [4].

B. QUANTIZATION AND PRUNING OPTIMIZATION STRATEGIES

Quantization and pruning are the main methods to achieve model compression, sacrificing a little accuracy for efficiency improvement to adapt to the limited resources of mobile devices. Quantization optimization reduces the model weights from 32-bit floating-point numbers (FP) to 32-bit floating-point numbers (FP). 32) Converting to 8-bit integers (INT8), 4-bit integers (INT4), or even smaller 2-bit integers (INT2) reduces model size by 75% while also increasing computational cost, making inference 3-4 times faster. When quantization leads to a decrease in accuracy, Quantization-Aware Training (QAT) is used, which simulates quantization error and adjusts model parameters during training to adapt to low-precision calculations. Pruning optimization is divided into structured pruning and unstructured pruning: Structured pruning refers to deleting entire channels, layers, or attention heads, such as deleting attention heads and network layers with a contribution of less than 5%. This compressed model can be deployed and used without a special inference framework. Unstructured pruning refers to deleting individual redundant weights (such as weights close to 0), which can achieve more... High compression ratio, but requires optimization using sparse matrix calculations to improve inference speed. Quantization and pruning are combined: first, pruning removes redundant structures, then quantization is performed. For example, pruning compresses by 3 times + INT8 quantization compresses by 4 times, ultimately achieving a 12-fold model volume compression while keeping accuracy loss within 3%.

C. INFERENCE ACCELERATION ADAPTATION STRATEGY

Inference acceleration adaptation targets three parts: framework optimization, hardware adaptation, and operator customization, maximizing the potential of mobile hardware to improve real-time performance. Model conversion and optimization are performed based on edge inference frameworks such as TensorFlow Lite and PyTorchMobile, utilizing model optimization tools provided by these frameworks (e.g., quantization and pruning functions of TFLiteConverter to remove some training data). Nodes not used during training, such as Dropout layers, are used to generate inference models suitable for mobile devices. The framework's graphics optimization options are enabled, leveraging its automatic fusion of operators like convolution, activation, and batch normalization to reduce data read/write latency during inference. Operators are customized based on the characteristics of different computing units on mobile devices, such as CPUs, GPUs, and NPUs. For self-attention operators on the CPU, a caching mechanism is used to reduce memory access overhead. Parallel computing logic is designed for the GPU, utilizing the advantages of multi-core parallel computing to improve the speed of matrix multiplication operations. The data format of operators is adapted to NPU (NHWC) to leverage the NPU's hardware acceleration capabilities in low-precision computing. On NPU-enabled devices, the inference speed of lightweight models accelerated by the NPU is faster than that of models inferred solely by the CPU. The processing speed is 5-8 times faster. Additionally, model sharding technology is used to break large models down into smaller shards, which are loaded into memory as needed, avoiding inference interruptions due to insufficient memory and improving deployment stability.

D. RESOURCE BALANCING STRATEGY

The resource balancing strategy balances model size, inference speed, and power consumption, adapting to different devices and dynamically allocating resources. Different lightweight versions are designed based on the hardware capabilities of different mobile devices (memory size, computing power, battery capacity). High-end devices (8GB RAM + flagship CPU, high-precision version: 6-layer encoder + 6-layer decoder + 6-head attention); mid-range devices (4GB~6GB RAM + mid-range CPU, balanced version: 4-layer encoder + 4-layer decoder + 4-head attention); low-end devices (2GB~4GB RAM + entry-level...) CPU, High-Efficiency Version: 2-layer encoder + 2-layer decoder + 2-head attention. Prioritizing real-time performance for low-end devices, the High-Efficiency Version is used; for mid-range devices balancing accuracy and speed, the Balanced Version is used; and for high-end devices ensuring translation quality, the High-Precision Version is used. Translation accuracy and inference speed are dynamically adjusted. Users can choose "High-Precision Mode" for formal document translation or "High-Speed Mode" for dialogue, menu, and other scenarios. The system automatically switches based on battery level — when the battery is below 20%, it automatically switches to High-Speed Mode to conserve energy. Improved memory management employs memory reuse technology, using the

same memory blocks to store intermediate features during inference, reducing peak memory usage; and an on-demand allocation strategy is adopted, allocating memory only to operators when necessary. To prevent fragmentation, lightweight Transformer technology allocates memory and releases its occupied space resources immediately after inference, ensuring smooth operation on low-end devices.

V. CONCLUSION

Lightweight Transformer technology provides an effective solution for real-time text translation on mobile devices, effectively addressing the adaptation issues of traditional models to mobile platforms. This paper analyzes the current research and application status, identifying current research achievements and shortcomings; elucidates its significance in user experience, practical application, industry communication, and data security; proposes improvement methods to optimize this technology; and provides a practical application path. With future development, advancements in model compression, hardware acceleration, and inference technologies will lead to significant improvements in translation accuracy, real-time performance, and resource consumption, enabling real-time mobile translation to move towards a "more accurate, faster, and better experience," thus playing a more stable role in global multilingual communication.

REFERENCES

[1] Y. Liu, “ Translation paratext and the construction of the literary image of Hong Kong's ‘People's Literature’,” Chinese Literature, no. 6, pp. 85–92, 2024.
[2] L. Wu, C. Ma, Y. Zhou, et al., “Research on text-image translation incorporating confidence,” J. Chinese Inf. Process., vol. 38, no. 12, pp. 64–73, 2024.
[3] Z. Huang and R. Liu, “Translation of pharmaceutical vocabulary and sentences under the guidance of text type theory,” Overseas English, no. 23, pp. 27–29, 2024.
[4] X. Hu and R. Liu, “ Translation of health science texts from the ‘ three-dimensional ’ transformation perspective of ecological translatology, ” English Square, no. 35, pp. 15 – 18, 2024, doi: 10.16723/j.cnki.yygc.2024.35.002.