Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 18 No. 22 (2024) | OPEN ACCESS

https://doi.org/10.3991/ijim.v18i22.52451

PAPER

Optimizing Offline Mode and Data Synchronization Techniques for Literature Translation Applications on Mobile Devices

Fei Wang[1](✉), **Yuanyuan Wang**[2]

[1]School of Humanities, Shangluo University, Shangluo, China

[2]School of Electronic Information and Electrical Engineering, Shangluo University, Shangluo, China

227047@slxy.edu.cn

ABSTRACT

The widespread use of mobile devices has intensified the demand for literature translation in both academic research and personal learning. However, existing translation applications face significant challenges regarding offline mode and data synchronization. Ensuring translation quality and efficient data synchronization, particularly under unstable or disconnected network conditions, has emerged as a critical issue. Current research predominantly focuses on local storage and update mechanisms in offline mode. However, limitations in storage capacity and update strategies often hinder translation effectiveness. Additionally, data synchronization techniques have been primarily studied in stable network environments, with insufficient attention to strategies that address poor network quality or intermittent connectivity. To address these gaps, this study explores the offline mode and data synchronization technologies in literature translation applications on mobile devices. The study proposes optimized storage and update strategies for the offline mode while enhancing the efficiency and reliability of data synchronization, ultimately improving the usability and user experience of translation applications.

KEYWORDS

mobile devices, literature translation, offline mode, data synchronization, user experience

1 INTRODUCTION

With the widespread adoption of mobile devices and the rapid advancement of technology, literature translation applications have increasingly become essential tools for academic research and daily learning [1–3]. However, these applications face numerous challenges in terms of offline mode and data synchronization [4–6]. In mobile environments, users frequently encounter unstable or disconnected network conditions, placing higher demands on the offline capabilities of translation applications [7–11]. Additionally, ensuring translation quality while enabling

Wang, F., Wang, Y. (2024). Optimizing Offline Mode and Data Synchronization Techniques for Literature Translation Applications on Mobile Devices. *International Journal of Interactive Mobile Technologies (iJIM)*, 18(22), pp. 115–129. https://doi.org/10.3991/ijim.v18i22.52451

Article submitted 2024-08-03. Revision uploaded 2024-09-14. Final acceptance 2024-09-18.

effective data synchronization in offline mode has become a pressing issue that requires urgent attention.

Research into offline modes and data synchronization for literature translation apps is crucial both theoretically and practically. These advancements are expected to boost translation efficiency across various environments, enhancing the dissemination and use of academic resources [12, 13]. Furthermore, improving offline modes and synchronization can greatly enhance the user experience, encouraging wider use and development of these applications [14–17]. Therefore, thorough research in these areas not only addresses current technological gaps but also supports the continued growth of the field.

Current studies identify several limitations. Offline modes often depend on local storage, which may be limited in capacity and have inefficient update mechanisms, leading to poor translation performance [18, 19]. Additionally, studies on data synchronization typically assume stable network conditions without addressing challenges posed by poor connectivity [20, 21]. These challenges limit the effectiveness of translation apps, highlighting the need for innovative research approaches to overcome these challenges.

This study primarily investigates the offline mode and data synchronization technologies of literature translation applications on mobile devices. First, the study thoroughly analyzes methods for implementing the offline mode, including strategies for data storage, updates, and access in offline environments. Second, this study explores the optimized solutions for data synchronization, with a particular focus on addressing synchronization challenges under varying network conditions. Through this study, it aims to propose an improved solution that enhances both the practicality and user experience of translation applications, contributing valuable insights and practical experience to the development of related technologies.

2 OFFLINE MODE FOR LITERATURE TRANSLATION APPLICATIONS ON MOBILE DEVICES

Implementing offline translation for literature on mobile devices presents several challenges, particularly when using the traditional Transformer model. First, the Transformer architecture involves a significant amount of redundant computation, primarily due to its multi-layered self-attention mechanisms and feedforward neural networks. This results in an extensive number of model parameters and a high demand for computational resources. However, mobile devices are typically constrained by limited hardware resources, such as insufficient computational power and storage space, making the deployment of a full-scale Transformer model difficult. Additionally, while the attention mechanism of the Transformer performs exceptionally well in handling long texts, it is relatively weak in extracting local features, which is particularly crucial in literature translation. Literature often contains a large number of specialized terms and context-dependent details, requiring precise comprehension of local features in offline translation. The limited ability of the Transformer to extract these features may lead to suboptimal translation performance. Furthermore, Transformer models in offline environments lack the capacity for dynamic updates, making it challenging to adapt to the constantly evolving linguistic characteristics within specific fields of literature. Offline mode requires fast and accurate translation on-device, but the computational complexity and redundancy of the Transformer model hinder its ability to meet these demands. Therefore, key challenges to be addressed in the offline mode of literature translation applications on mobile devices include optimizing the Transformer structure, enhancing its local feature extraction capabilities, and reducing its dependency on computational resources.

Several optimization methods can be employed to overcome these challenges: (a) A lightweight attention mechanism can be developed by dividing the traditional attention mechanism into two parts. One part focuses on global information capture by using the attention mechanism to handle long-range dependencies, and the other utilizes dynamic convolutions to capture local information, thereby improving the handling of specialized terms and contextual details. This dual-branch mechanism, incorporating a gating mechanism to fuse global and local features, can replace the traditional attention mechanism and feedforward neural networks, reducing model complexity and computational overhead. (b) To alleviate redundancy between layers, a cross-layer parameter sharing strategy can be introduced. By grouping layers and sharing parameters within the same group, the frequency of parameter updates can be increased, reducing redundant computations and improving the operational efficiency of the model on mobile devices.

2.1 Lightweight convolution

In implementing offline literature translation on mobile devices, the application of deep convolution in the Transformer model can significantly improve the efficiency of feature extraction. Deep convolution, as an efficient feature extractor, decomposes traditional convolution operations to maintain low computational complexity while extracting more precise features. This advantage becomes particularly evident in literature translation tasks on mobile devices. Specifically, deep convolution processes input features using multiple convolutional kernels, with each kernel operating on a single channel. This design allows each kernel to focus on extracting local features from specific channels, thereby enhancing the capture of local information. For instance, when processing sentences in literature texts, deep convolution can capture the local features of sentences using a sliding window. After the convolution operation, pointwise convolution can be applied to fuse features from different channels, generating feature representations similar to traditional convolutions. The formula for deep convolution is as follows:

$$DC\left(A, Q_{Z,:}, u, z\right) = \sum_{k=1}^{j} Q_{z,k} * A_{\left(u+k-\left\lceil \frac{j+1}{2} \right\rceil\right),z} \tag{1}$$

Deep convolution significantly reduces the number of parameters compared to traditional convolution. Let f represent the input dimension and j the size of the convolution kernel. The number of parameters can thus be reduced from $f^2 j$ to fj.

Parameter compression and computational optimization are crucial for mobile applications. Given the limited resources of devices, lightweight convolution can significantly enhance operational efficiency while maintaining model performance. Specifically, lightweight convolution employs a fixed context window and weights that do not change over time steps, greatly reducing the number of parameters and computational complexity. In literature translation tasks, lightweight convolution reduces the number of required parameters by dividing the convolutional kernels into multiple groups, with each group using the same kernel. For example, if lightweight convolution is divided into 16 groups, the number of parameters decreases from 3,584 in deep convolution to 112. This not only reduces the computational burden but also lowers storage requirements. Assuming that the number of channels per head in multi-head weight sharing is represented by f/G, the calculation for the output channel z and the u-th element in lightweight convolution can be expressed as follows:

$$LC\left(A, Q_{\left[\frac{zG}{f}\right],:}, u, z\right) = DC\left(A, SM\left(Q_{\left[\frac{zg}{f}\right],:}, u, z\right)\right) \tag{2}$$

2.2 Lightweight attention mechanism module

To address the challenge of redundant computations within the Transformer model, a lightweight attention mechanism was proposed in this study to replace the traditional attention mechanism and feedforward neural networks in the Transformer. Figure 1 illustrates the structural diagram of the lightweight attention mechanism model.

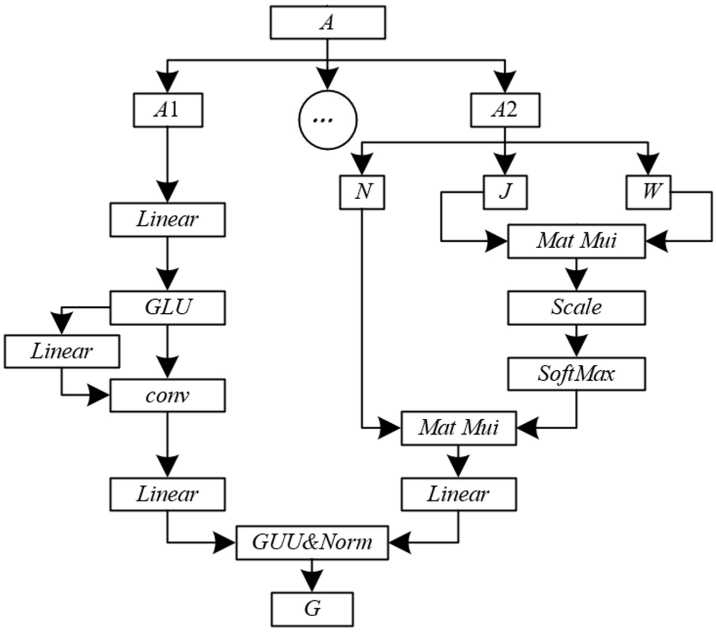

Fig. 1. Structural diagram of the lightweight attention mechanism model

The design of the proposed lightweight attention mechanism incorporates a dual-branch structure. This structure employs both dynamic convolution and the traditional attention mechanism, with the aim of optimizing the capture of local and global features. While the traditional attention mechanism excels at capturing global information, it is relatively less effective in extracting local features, particularly when handling literature that involves complex contexts and specialized terminology. By introducing dynamic convolutional networks, the lightweight attention mechanism enhances the ability to capture local features, improving comprehension of detailed information in literature. The dynamic convolutional networks offer strong feature-capturing capabilities with a relatively small number of parameters, allowing the lightweight attention mechanism to maintain a lower computational burden and reduced storage requirements, which is crucial in resource-constrained mobile environments.

Specifically, in the lightweight attention mechanism model, the input sentences from the literature are split into two equal parts: A_1 and A_2, where $f1 = f2$, and $f = f1 + f2$. This splitting operation ensures that dynamic convolution and the attention mechanism can simultaneously process and extract features from the input data. In the dual-branch structure, one branch employs the traditional attention mechanism to capture global information, while the other branch utilizes dynamic convolutional networks focused on capturing local features. To remain compatible with the multi-head attention mechanism design of the Transformer, the lightweight attention mechanism sets the number of heads for both the attention mechanism and dynamic convolution to G, with the dimension of each head set to $f_{MO} = f/2G$. This setup allows each branch to efficiently extract and process features within its respective feature space. The extracted features from both methods are then fused through a gating mechanism, ensuring that the lightweight attention mechanism can achieve stronger feature extraction with fewer parameters, particularly in resource-constrained mobile device environments. The formula for the output of the traditional attention layer on the left branch is given below:

$$AT(W, J, N) = SM\left(\frac{WJ^S}{\sqrt{f_{MO}}}\right)N \tag{3}$$

Assuming that W, J, and N are the parameters obtained from multiplying A_1 with Q_w, Q_j, and Q_n, which are the learnable parameters.

To further enhance the ability to capture local features, a dynamic convolution design was adopted in the right branch. This design aims to strengthen the extraction of local features. Similar to lightweight convolution, dynamic convolution first expands the input feature dimension from f to $2f$ through a linear layer, after which the features are re-fused to dimension f via a gating mechanism. This process is analogous to the $W \times J^T$ operation in the attention mechanism. Subsequently, the feature output of the sentence is fed into the dynamic convolution layer.

Dynamic convolution introduces a time-step-dependent convolutional kernel, *dynamic weight*, which differs from lightweight convolution. This approach allows the model to dynamically adjust the convolutional kernel's weights at different time steps, thereby capturing local features with greater precision. Therefore, dynamic convolution can adapt to contextual changes in input sentences, improving the understanding of sentence details and complex contexts in literature. Assuming that the learnable linear parameter is represented by $Q^{w}_{g,k,z}$, the dynamic convolution formula is given as follows:

$$DC(A,u,z) = LC(A,d(A_u)_{g:},u,z) \tag{4}$$

$$d(A_u) = \sum_{z=1}^{f} Q^{W}_{g,k,z} A_{u,z} \tag{5}$$

Furthermore, the lightweight attention mechanism introduces a dual-branch structure and a gating mechanism, aimed at optimizing computational efficiency and improving feature extraction capability. Although the feedforward neural networks in the traditional Transformer model enhance the expressive power of the model, the large number of parameters and complexity pose significant challenges for mobile devices. The feedforward networks typically require expanding the dimension by fourfold, thus introducing a substantial number of parameters. In the lightweight attention mechanism, the dual-branch structure splits the input encoding into two equal parts, A_1 and A_2, which are processed separately by dynamic convolution and the attention mechanism. This setup allows dynamic convolution to capture local features while the attention mechanism handles global information. To replace the feedforward neural networks, the lightweight attention mechanism incorporates a gated recurrent unit (GRU) after the dual branches. This structure performs a similar nonlinear fusion function as the feedforward networks but significantly reduces model complexity and the number of parameters. The GRU not only lowers computational costs but also avoids the computational bottleneck caused by the large number of parameters in traditional feedforward networks. As a result, the model can perform literature translation more efficiently on resource-constrained mobile devices, ensuring a balance between translation quality and operational efficiency. Assuming the learnable linear transformation is represented by Q_h, concatenation is represented by $[\cdot;\cdot]$, the activation function is denoted as $\delta(\cdot)$, and $\odot$ represents element-wise multiplication, the GRU is defined by the following equations:

$$c = \delta(Q_h[AT(\cdot); DC(\cdot)]) \tag{6}$$

$$g = (1 - c)\Phi AT(\cdot) + c\Phi DC(\cdot) \tag{7}$$

To ensure that the mean and variance of the output from the gating mechanism remain the same, the layer normalization operation is required, as shown in the following formula:

$$G = (LN(g)) \tag{8}$$

3 DATA SYNCHRONIZATION FOR LITERATURE TRANSLATION APPLICATIONS ON MOBILE DEVICES

Data synchronization is a crucial process in ensuring the consistency of translation progress and results in mobile literature translation applications. To optimize the synchronization process while reducing resource consumption and improving efficiency, a feature-value-based encryption synchronization scheme was proposed. This approach involves extracting feature values from file content for message locking encryption, replacing the traditional hash-based method. The feature-value encryption synchronization offers several advantages, particularly by mitigating the issue of incremental amplification. Since the encryption is based on specific features extracted from the file content, rather than relying on the hash of the entire file, only the modified portions of the file require synchronization when minor changes occur. This reduces both data transmission volume and computational overhead. Moreover, considering the resource constraints of mobile devices, the feature-value-based encryption synchronization scheme significantly enhances cloud synchronization efficiency and lowers the burden on data transmission and processing, ensuring efficient synchronization of literature even under limited bandwidth and computational capacity.

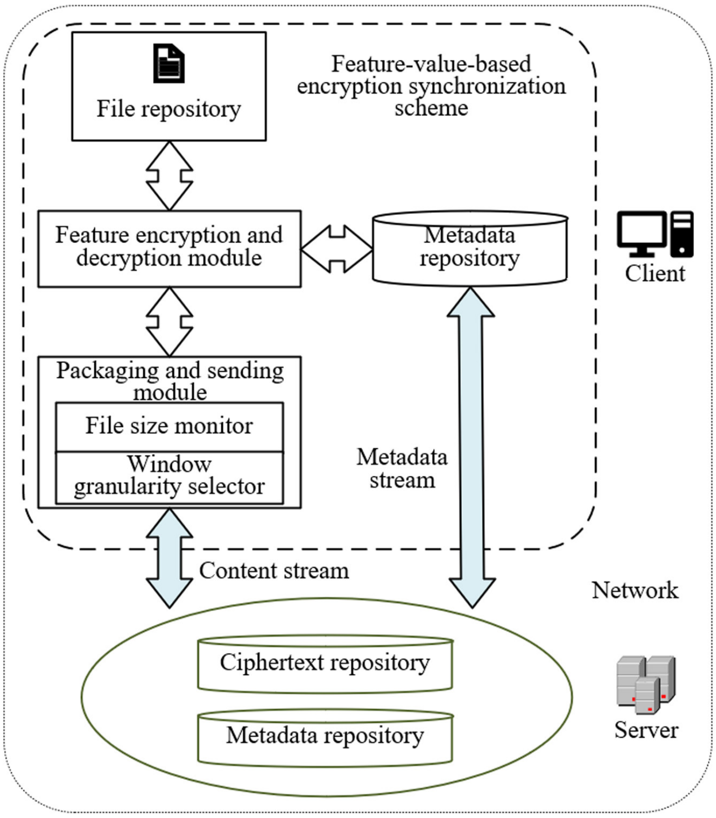

Fig. 2. Overall framework of the feature-value-based encryption synchronization scheme

Figure 2 illustrates the overall framework of the feature-value-based encryption synchronization scheme. This scheme includes two key modules: the feature encryption and decryption module and the packaging and sending module, both of which play critical roles. The feature encryption and decryption module are responsible for calculating the Rabin fingerprints of each file data block using a sliding window and selecting the representative feature values from these fingerprints. This process aims to reduce redundant computation, ensuring that only the most representative feature values are used for file encryption, thereby improving encryption efficiency and reducing the consumption of computational resources on the device. The packaging and sending module primarily package the encrypted file ciphertext along with the key metadata required for decryption. It also monitors the properties of the package to select the appropriate sliding window granularity for incremental synchronization. This design is particularly suitable for offline literature translation applications on mobile devices, as the granularity of incremental synchronization can be flexibly adjusted. This ensures that only the changed portions of the file data are transmitted under bandwidth-constrained conditions, thereby significantly improving the efficiency of data synchronization. Furthermore, by packaging and encrypting the key metadata, the packaging and sending module enhances the security of data transmission, ensuring that the content of the literature remains secure during synchronization.

In the context of literature translation applications on mobile devices, feature-value-based encryption optimizes data synchronization and encryption efficiency by generating fuzzy hash values rather than exact ones. The core idea is to use feature values to represent the characteristics of the file content, enabling efficient encryption and synchronization. Specifically, when minor changes occur in the file content, traditional exact hash values would change significantly, requiring the generation of new keys and re-encryption of the file, thereby increasing the complexity of key management. In contrast, the use of Sketch hashing technology generates feature values for file blocks, where blocks with identical Sketch hash values are likely to be similar or nearly duplicated in content. This characteristic allows the feature values to remain stable even when minor insertions or modifications occur in the file content, thus eliminating the need for frequent key updates.

Specifically, the feature-value-based encryption synchronization scheme utilizes a sliding window of 48 bytes, which moves across the file byte by byte. During this process, Rabin fingerprints are calculated for each window. As the sliding window progresses, the synchronization scheme continuously updates the maximum values of these fingerprints, ensuring that the most representative feature values from the entire file content are identified. This method allows the synchronization scheme to effectively detect changes within the file and apply encryption to specific block boundaries. The primary advantage of this approach lies in its ability to encrypt and synchronize only the modified portions when minor changes occur in the file without reprocessing the entire file. This significantly reduces both the volume of data transmission and the computational overhead. By reusing the computed Rabin fingerprints, the sliding window can rapidly identify feature values as it traverses the file, accelerating the feature-search process. Once the entire file has been scanned and the fingerprints calculated, the synchronization scheme selects the maximum fingerprint value as the feature value for the file and uses it as the key to encrypt all blocks uniformly.

Figure 3 illustrates the synchronization process of the proposed scheme. The data synchronization process is described below through a specific example:

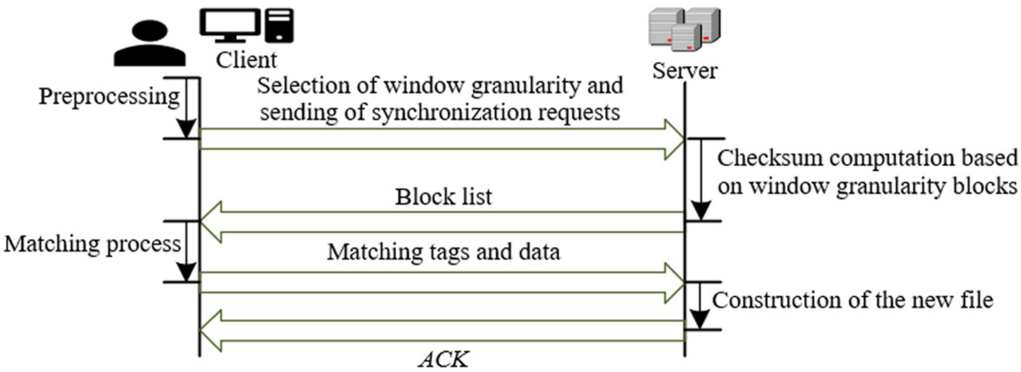

Fig. 3. Synchronization process diagram of the proposed scheme

a) On mobile devices, users may edit and update multiple translated documents, such as adding new translation paragraphs, modifying existing content, or inserting annotations. These changes are saved in a local folder. The client application first scans this folder, recursively identifying all modified documents and adding them to the pending transmission package. This ensures that all modified documents are included in the synchronization process to be updated to the cloud server.

b) For each modified translation document, the client applies a 48-byte sliding window that moves through the entire file byte by byte. During this step, the client calculates the Rabin fingerprints for each window and records the maximum fingerprint value. This feature value serves as a unique identifier for the document, enabling the client to efficiently identify the modified portions. If the file is long or contains many changes, the calculation of the feature value effectively distinguishes different versions, thus avoiding redundant data transmission during synchronization.

c) During the synchronization process, the client uses the document's feature value as a key to encrypt the translation file into ciphertext before synchronization. This step ensures the security of the data during transmission. Since the content of the literature may include sensitive or copyright-protected information, the encryption process is a crucial measure for protecting user privacy and data security. If the client identifies that the file is small, the system adopts a finer 700-byte window granularity for incremental synchronization, reducing data transmission volume and improving synchronization efficiency.

d) The client sends a synchronization request containing the fine-grained window information to the server. Upon receiving this request, the server processes the literature data on its side using the same 700-byte window granularity. The server then divides the literature into several blocks, computes the hash value for each block, and returns this information to the client. This process ensures that the client can accurately compare and update the literature data.

e) Once the client receives the block list from the server, it compares the ciphertext in package P with the block list stored on the server. This step is used to verify whether the ciphertext blocks uploaded by the client match the file blocks stored on the server. For literature translation applications, this verification process ensures the correct synchronization of translated content, preventing data loss or errors. If any mismatched blocks are identified, the client marks these blocks for resynchronization or further processing.

f) After verification is complete, the client updates the successfully synchronized file content within the local translation application while also ensuring that the server holds the most up-to-date literature data. This process guarantees that translation progress and content remain consistent across different devices, with all modifications accurately synchronized. The client then updates its status and notifies the user of the synchronization results, enabling the user to continue the translation work efficiently.

4 EXPERIMENTAL RESULTS AND ANALYSIS

In the experimental results presented in Figure 4, significant differences in Bilingual Evaluation Understudy (BLEU)-4 scores can be observed across various translation methods during iterations 30 to 39. As the number of iterations increases, the BLEU-4 score of ALBERT, which means A Lite BERT (Bidirectional Encoder Representations from Transformers), improves gradually from 21.6 to 30, though slight fluctuations can be observed after the 35th iteration, with a peak score of 30. In contrast, Distilled BERT (DistilBERT) shows a steady improvement in BLEU-4 score, increasing from 23.6 to 29.7 between iterations 30 and 39. The performance of TinyBERT mirrors that of DistilBERT. However, after the 36th iteration, TinyBERT has a slightly higher BLEU-4 score (30.8) than DistilBERT, though it has a slight decline in subsequent iterations. The proposed method outperforms all other methods, starting with a BLEU-4 score of 26.8 at iteration 30 and reaching a peak of 31.66. Although minor fluctuations occur in the following iterations, the BLEU-4 score remains consistently high.

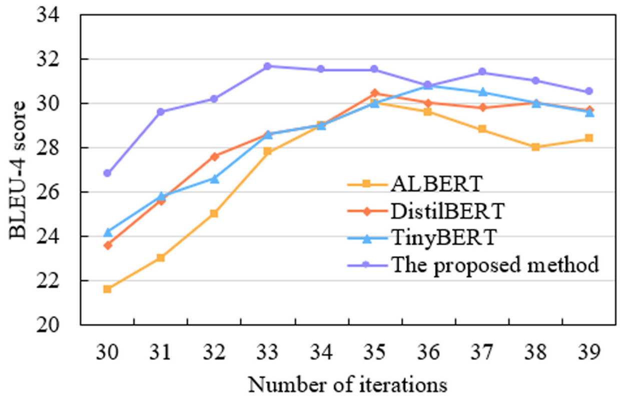

Fig. 4. Comparison of BLEU-4 scores for different translation methods

In Table 1, the BLEU-4 scores of the proposed translation method demonstrate varying performance across different datasets. The BLEU-4 score for the parallel corpus is 29.12, slightly increasing to 30.14 for the pseudo-parallel corpus. The BLEU-4 score for the monolingual corpus shows a significant improvement, reaching 42.23, while the pseudo monolingual corpus has an even higher BLEU-4 value of 44.32. In addition to the changes in BLEU-4 scores, the number of model parameters also varies. The pseudo-monolingual corpus has the fewest parameters at 6.51 M, while the other datasets exhibit relatively higher parameter counts: 6.98 M for the parallel corpus, 6.78 M for the pseudo-parallel corpus, and 6.77 M for the monolingual corpus. From these results, it can be concluded that the pseudo-monolingual corpus has the highest BLEU-4 score, indicating that this dataset has the most beneficial effect on model training. This could be attributed to the closer alignment of the pseudo-monolingual corpus content with the target language, allowing the model to better learn linguistic features. In contrast, the parallel and pseudo-parallel corpora display relatively lower BLEU-4 scores, which may suggest that these datasets lack sufficient translation quality or diversity, thereby impacting model performance. The monolingual corpus performs better than the parallel corpus but slightly underperforms compared to the pseudo-monolingual corpus. This indicates that while the monolingual dataset offers improvements to the model, it still lags behind the pseudo-monolingual corpus in effectiveness.

Table 1. Comparative analysis of the proposed translation method on different datasets

Dataset	BLEU-4	Parameters
Parallel corpus	29.12	6.98 M
Pseudo-parallel corpus	30.14	6.78 M
Monolingual corpus	42.23	6.77 M
Pseudo-monolingual corpus	44.32	6.51 M

Table 2 compares the translation results across different datasets. For Sentence 1, the translations from the parallel corpus and monolingual corpus are relatively similar, with the monolingual corpus translation fully matching the reference translation. In contrast, the parallel corpus translation contains “<unk>” tokens, indicating an inability to recognize certain words. The reference translation provides an accurate rendering of the original content. For Sentence 2, the translation from the monolingual corpus also fully aligns with the reference translation, while the parallel corpus translation again contains “<unk>” tokens, leading to information loss. These differences highlight the varying performance of different datasets in handling specific translation tasks. The analysis reveals that the monolingual corpus outperforms the parallel corpus in terms of translation quality. The monolingual corpus provides more accurate translations without the occurrence of “<unk>” tokens, indicating that it offers richer and more consistent linguistic information for model training. In contrast, the parallel corpus suffers from missing vocabulary, which results in a decline in translation quality and incomplete translations. Overall, the monolingual corpus demonstrates a clear advantage in ensuring translation accuracy and fluency, while the parallel corpus may be hindered by insufficient data when handling specific terms.

Table 2. Comparative translation analysis

Sentence Type	Sentence Content
Sentence 1	*Milk is well known for its rich nutritional content and effectively provides most of the nutrients required by the human body. Its easy accessibility means that most people can benefit from the nutritional supply of milk to meet their bodily needs.*
Reference translation	众所周知，牛奶含有丰富的营养成分，能够有效补充人体所需大部分营养。其易于获取的特性使得大部分人都能够享受到牛奶带来的营养补给，以满足身体的需求。
Parallel corpus	牛奶因其丰富的<*unk*>成分而广受欢迎，能够有效提供人体所需的大部分<*unk*>。其易于获取的特点使得大多数人都能够从牛奶的营养补给中受益，满足身体的需求。
Monolingual corpus	牛奶因其丰富的营养成分而广受欢迎，能够有效提供人体所需的大部分营养。其易于获取的特点使得大多数人都能够从牛奶的营养补给中受益，满足身体的需求。
Sentence 2	*With the arrival of the fishing season, the variety and abundance of seafood are continuously increasing.*
Reference translation	随着捕鱼季节的到来，海鲜的种类与丰富度都在不断增加。
Parallel corpus	随着<*unk*>季节的到来，海鲜的种类和丰富度不断<*unk*>。
Monolingual corpus	随着捕鱼季节的到来，海鲜的种类和丰富度不断增加。

Table 3. Experimental results of different models on three metrics

Dataset	Model	BLEU-4	Parameters	Speed
Parallel corpus	Traditional Transformer	28.89	18.23 M	× 1.0
	Proposed method	30.02	6.78 M	× 0.78
	DistilBERT	30.23	6.84 M	× 0.88
	TinyBERT	30.85	5.89 M	× 0.61
	ALBERT	31.65	5.98 M	× 0.68
Monolingual corpus	Traditional Transformer	42.36	20.33 M	× 1.0
	Proposed method	44.25	6.58 M	× 0.77
	DistilBERT	44.21	6.81 M	× 0.81
	TinyBERT	44.23	5.68 M	× 0.62
	ALBERT	45.22	5.95 M	× 0.64

Table 3 presents the performance of different models across the parallel and monolingual corpora. For the parallel corpus dataset, the ALBERT model achieves the highest BLEU-4 score of 31.65, while TinyBERT scores 30.85, the proposed method 30.02, DistilBERT 30.23, and the traditional Transformer 28.89. In the monolingual corpus dataset, ALBERT again demonstrates the best performance, with a BLEU-4 score of 45.22, followed by the proposed method at 44.25, TinyBERT at 44.23, DistilBERT at 44.21, and the traditional Transformer at 42.36. In terms of parameter count and speed, ALBERT generally exhibits fewer parameters and faster processing speed. The proposed method, DistilBERT, and TinyBERT also show a reasonable balance in parameter count, with speed performance across all models being relatively high.

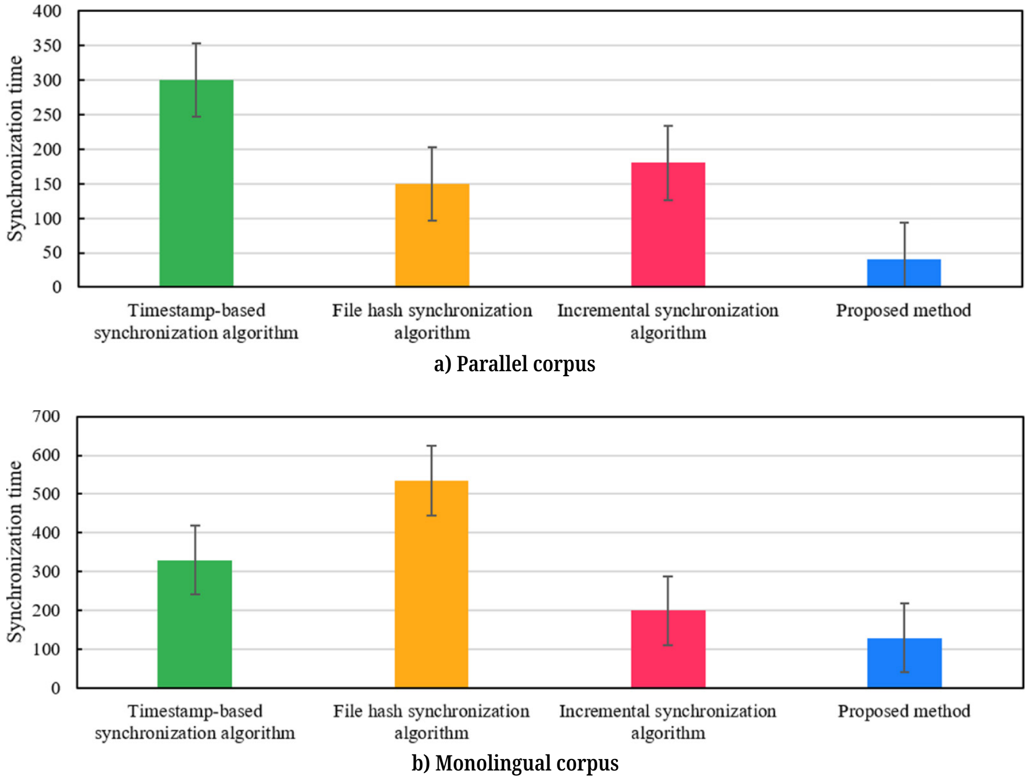

Fig. 5. Synchronization time for different data synchronization schemes driven by different corpora

In Figure 5, the synchronization times of different data synchronization algorithms on the parallel and monolingual corpora are presented. For the parallel corpus, the synchronization time for the timestamp-based synchronization algorithm is 300 seconds, the file hash synchronization algorithm takes 150 seconds, the incremental synchronization algorithm takes 180 seconds, and the proposed method completes synchronization in 40 seconds. For the monolingual corpus, the synchronization time for the timestamp-based synchronization algorithm is 330 seconds, the file hash synchronization algorithm takes 535 seconds, the incremental synchronization algorithm takes 200 seconds, and the proposed method takes 130 seconds. These results indicate that the proposed method significantly reduces synchronization time on both datasets compared to the other algorithms. The analysis shows that the proposed method exhibits clear advantages in synchronization time for both the parallel and monolingual corpora, particularly for the monolingual corpus, where the synchronization time of 130 seconds is markedly lower than that of other algorithms. This suggests that the proposed method offers higher efficiency and superior performance when handling large-scale data. In contrast, the file hash synchronization algorithm has the shortest synchronization time for the parallel corpus but performs significantly slower on the monolingual corpus. This highlights the performance differences of different algorithms across datasets, which may be influenced by the structure and scale of the data. Overall, the proposed method consistently provides faster synchronization speeds across multiple datasets, demonstrating its efficiency in practical applications.

5 CONCLUSION

This study primarily focuses on literature translation applications on mobile devices, particularly optimizing offline mode and data synchronization technologies. By comparing BLEU-4 scores across different translation methods, the proposed method demonstrates excellent translation quality, especially in handling monolingual corpora, outperforming the traditional model and other modern models. Based on the experimental results of various translation methods, the ALBERT model shows the best performance in the BLEU-4 score, highlighting its advantages in translation accuracy. This study also explores data synchronization technologies, including synchronization time comparisons between the timestamp-based, file hash, and incremental synchronization methods and the proposed method. The results indicate that the proposed method has significant advantages in synchronization speed, particularly when processing large-scale data.

In summary, the value of this study lies in providing an efficient and accurate offline translation and data synchronization solution that is well-suited for translation applications on mobile devices. However, the limitations of the study include the potential impact of the scale and diversity of sample data on the general applicability of the results, as well as the need for further verification of synchronization efficiency under different network conditions. Future research directions could involve investigating the performance of algorithms in more complex network environments and expanding the scope to more language pairs and datasets to validate and optimize both translation quality and synchronization efficiency. Additionally, exploring the integration of machine learning and deep learning technologies could further enhance the intelligence of translation and synchronization.

6 ACKNOWLEDGEMENT

This paper was supported by the Education Department of Shaanxi Province of China (Grant No.: 21JK0110) and Shangluo University (Grant No.: 20SLWH07).

7 REFERENCES

[1] R. S. Al-Maroof, S. A. Salloum, A. Q. AlHamadand, and K. Shaalan, “Understanding an extension technology acceptance model of Google Translation: A multi-cultural study in United Arab Emirates,” *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 14, no. 3, pp. 157–178, 2020. https://doi.org/10.3991/ijim.v14i03.11110

[2] M. Kabát, “Possible translation problems, their causes, and solutions in agile localization of software,” *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 17, no. 1, pp. 129–140, 2023. https://doi.org/10.3991/ijim.v17i01.36367

[3] P. Nayak, R. Haque, J. D. Kelleher, and A. Way, "Investigating contextual influence in document-level translation," *Information*, vol. 13, no. 5, p. 249, 2022. https://doi.org/10.3390/info13050249

[4] X. Tan, L. Y. Zhang, and G. D. Zhou, "Document-level neural machine translation with hierarchical modeling of global context," *Journal of Computer Science and Technology*, vol. 37, pp. 295–308, 2022. https://doi.org/10.1007/s11390-021-0286-3

[5] S. M. Fakhrahmad, M. H. Sadreddini, and M. Zolghadri Jahromi, "A proposed expert system for word sense disambiguation: Deductive ambiguity resolution based on data mining and forward chaining," *Expert Systems*, vol. 32, no. 2, pp. 178–191, 2015. https://doi.org/10.1111/exsy.12075

[6] S. H. Na, I. S. Kang, and J. H. Lee, "Parsimonious translation models for information retrieval," *Information Processing & Management*, vol. 43, no. 1, pp. 121–145, 2007. https://doi.org/10.1016/j.ipm.2006.04.005

[7] R. Aggarwal, S. Pandey, A. K. Tiwari, and G. Harit, "Survey of mathematical expression recognition for printed and handwritten documents," *IETE Technical Review*, vol. 39, no. 6, pp. 1245–1253, 2022. https://doi.org/10.1080/02564602.2021.2008277

[8] S. Singh, A. Sharma, and V. K. Chauhan, "Indic script family and its offline handwriting recognition for characters/digits and words: A comprehensive survey," *Artificial Intelligence Review*, vol. 56, no. S3, pp. 3003–3055, 2023. https://doi.org/10.1007/s10462-023-10597-y

[9] M. Sanaullah *et al.*, "A real-time automatic translation of text to sign language," *Computers, Materials & Continua*, vol. 70, no. 2, pp. 2471–2488, 2022. https://doi.org/10.32604/cmc.2022.019420

[10] S. P. H. Boroujeni and A. Razi, "IC-GAN: An improved conditional generative adversarial network for RGB-to-IR image translation with applications to forest fire monitoring," *Expert Systems with Applications*, vol. 238, p. 121962, 2024. https://doi.org/10.1016/j.eswa.2023.121962

[11] L. M. Patil, D. H. Parkinson, N. R. Zuniga, H. J. L. Lin, B. C. Naylor, and J. C. Price, "Combining offline high performance liquid chromatography fractionation of peptides and intact proteins to enhance proteome coverage in bottom-up proteomics," *Journal of Chromatography A*, vol. 1701, p. 464044, 2023. https://doi.org/10.1016/j.chroma.2023.464044

[12] L. Gao, Y. Ma, and P. Ma, "The development trend of English instant translation software in the background of mobile internet," *Applied Mathematics and Nonlinear Sciences*, vol. 9, no. 1, pp. 1–16, 2024. https://doi.org/10.2478/amns-2024-1710

[13] U. Bozuyuk, P. Wrede, E. Yildiz, and M. Sitti, "Roadmap for clinical translation of mobile microrobotics," *Advanced Materials*, vol. 36, no. 23, 2024. https://doi.org/10.1002/adma.202311462

[14] E. Rakun, S. Muzahidin, I. S. A. Darmana, and W. Setiaji, "SIBI (Sign System Indonesian Language) text-to-3D animation translation mobile application," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 9, pp. 440–450, 2022. https://doi.org/10.14569/IJACSA.2022.0130950

[15] L. M. Vortmann, P. Weidenbach, and F. Putze, "AtAwAR translate: Attention-aware language translation application in augmented reality for mobile phones," *Sensors*, vol. 22, no. 16, p. 6160, 2022. https://doi.org/10.3390/s22166160

[16] Y. Gao, "Design of English translation mobile information system based on recurrent neural network," *Mobile Information Systems*, vol. 2022, no. 1, 2022. https://doi.org/10.1155/2022/8053285

[17] L. Wang and H. Yang, "Mobile edge computing-based real-time English translation with 5G-driven network support," *International Journal of Distributed Systems and Technologies*, vol. 13, no. 2, pp. 1–14, 2022. https://doi.org/10.4018/IJDST.291078

[18] V. Dache, A. A. Năstase, I. A. Sgârciu, A. M. Anghel, and V. Sgârciu, "Experimental analysis of the pneumatic distributor with mobile translation armature and digital PWM control," *UPB Scientific Bulletin, Series C: Electrical Engineering and Computer Science*, vol. 84, no. 3, pp. 59–70, 2022.

[19] C. Andrade, L. F. Teixeira, M. J. M. Vasconcelos, and L. Rosado, "Data augmentation using adversarial image-to-image translation for the segmentation of mobile-acquired dermatological images," *Journal of Imaging*, vol. 7, no. 1, p. 2, 2020. https://doi.org/10.3390/jimaging7010002

[20] R. M. Kagalkar and S. V. Gumaste, "Mobile application-based translation of sign language to text description in Kannada language," *International Journal of Interactive Mobile Technologies (iJIM)*, vol. 12, no. 2, pp. 92–112, 2018. https://doi.org/10.3991/ijim.v12i2.8071

[21] G. E. Su, M. S. Sunar, and A. W. Ismail, "Device-based manipulation technique with separated control structures for 3D object translation and rotation in handheld mobile AR," *International Journal of Human-Computer Studies*, vol. 141, p. 102433, 2020. https://doi.org/10.1016/j.ijhcs.2020.102433

8 AUTHORS

Fei Wang graduated from Hong Kong Polytechnic University with the degree of Master of Arts and works at Shangluo University, focusing on the research directions of English language arts and applied translation (E-mail: 227047@slxy.edu.cn).

Yuanyuan Wang graduated from Xi'an Shiyou University of China with a master's degree and works at Shangluo University, focusing on the research directions of signal and signal processing (E-mail: 230030@slxy.edu.cn).