ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20231120001119
Original scientific paper

An Experimental Study on Improved Sequence-to-Sequence Model in Machine Translation

Yuan-shuai LAN*, Chuan LI, Xueqin MENG, Tao ZHENG, Mincong TANG

Abstract: This paper presents the N-Seq2Seq model for enhancing machine translation quality and efficiency. The core innovations include streamlined attention mechanisms for focusing on crucial details, word-level tokenization to preserve meaning, text candidate frames for prediction acceleration, and relative positional encoding reinforcing word associations. Comparative analyses on English-Chinese datasets demonstrate approximately 4 BLEU score improvements over baseline Seq2Seq and 2 BLEU gains over Transformer models. Moreover, the N-Seq2Seq model reduces average inference time by 60% and 43% respectively. These techniques improve contextual modeling, reduce non-essential information, and accelerate reasoning. Importantly, the model achieves higher accuracy with low overhead, making it possible to deploy on mobile applications, while the Chinese-centric design can also be quickly adapted to other languages.

Keywords: BLEU evaluation; machine translation; N-Seq2Seq model; Seq2Seq model; WoBert model

1 INTRODUCTION

In recent years, with the increase in international trade and global cooperation, machine translation (MT) has become increasingly important in eliminating language barriers and promoting cross-cultural communication. Neural Machine Translation (NMT) has emerged as the mainstream, replacing traditional Statistical Machine Translation (SMT) methods.

In the past few years, several important research directions have emerged in the field of MT. Lample et al. (2017) proposed an unsupervised MT method [1] that reduces the reliance on parallel corpora, but its translation quality still requires improvement. Vaswani et al. (2017) introduced the Transformer model [2] to better capture global information, but challenges exist in terms of hardware and data requirements. Devlin (2019) introduced the BERT model [3], which improved contextual understanding and semantic representation for MT tasks, but it necessitates significant computational resources. However, these models have limited adaptability in Chinese translation, with issues related to ambiguity and context dependency.

Despite significant progress brought to the MT field by previous research, the following issues still need to be addressed: (1) resolving Chinese translation ambiguities, (2) reducing computational resource requirements, (3) addressing the problem of repetitive computations, and (4) addressing limited context information [4].

In the translation process, it is easy to lead to unclear expressions when involving proper nouns. Take "Apple" as an example, the word can refer to both fruits and a cell phone. When the word appears but the context is not clear, such as "This apple is so expensive, it costs thousands of dollars to buy one", the translator may incorrectly interpret it as a fruit, while in fact it refers to an "apple cell phone", resulting in unclear positioning and ambiguity in the translation result. In addition, when translating longer sentences or short texts, there are cases where a noun is defined at the beginning of a sentence to represent a certain thing. As the translation proceeds, when the noun appears again in the later text, translation errors may occur, failing to fully understand the contextual context and leading to incorrect translation results. Moreover, the understanding of the context will also consume a lot of computational resources, while the model uses tens of billions of data during model training, which also adds an extra burden.

Translated with DeepL.com (free version): To address the aforementioned issues, this paper introduces a novel machine translation model, the N-Seq2Seq model. This model incorporates the WoBERT model and relative positional encoding to enhance the sequential learning of sentence components. Additionally, it utilizes a multi-head attention mechanism to improve inference and training speed, and introduces a text candidate frame to accelerate translation. Compared to traditional Seq2Seq models [5], the N-Seq2Seq model brings the following innovations: (1)Streamlined attention mechanisms: The N-Seq2Seq model employs two attention mechanisms, specifically focusing on global information and key details, achieved through hyperparameter tuning; (2) Word-level tokenization: This model employs word-level tokenization, enhancing text segmentation accuracy, ensuring the preservation of the original meaning in translated text; (3) Text candidate frame: A text candidate frame is introduced in the decoder to predict and accelerate the model's inference speed; (4) High performance and low resource demands: The N-Seq2Seq model outperforms traditional Transformer models in BLUE scores while demanding lower hardware resources, making it suitable for deployment on low-end devices. That is, the N-Seq2Seq model aims to improve the efficiency and quality of machine translation by combining the WoBERT model and relative positional coding with a multi-head attention mechanism and a text candidate framework, which achieves high performance and low resource requirements by using a simplified attention mechanism, word-level tokenization, and other innovations.

In summary, the N-Seq2Seq model represents a significant innovation in machine translation, with the potential to enhance translation quality and accelerate translation speed. This model excels in Chinese-based translation tasks and holds promise for adaptation to other languages [6]. Most importantly, the N-Seq2Seq model has relatively low computational resource requirements, making it suitable for low-resource environments such as mobile devices and opening up new possibilities for the development of the MT field.

2 RELATED WORKS

In recent years, the field of machine translation has made significant progress, with researchers employing various methods to address various translation challenges. From the early days of statistical machine translation (SMT) methods to traditional machine learning approaches, and now to today's deep learning-based Seq2Seq models, machine translation continues to evolve and become more intelligent.

Early statistical machine translation (SMT) methods laid the foundation for machine translation research, relying mainly on phrase models and statistical alignment models. For the first time, researchers used massively parallel corpora to construct statistical translation models by estimating translation probabilities, the most famous of which include the IBM model and phrase models [7].

As technology advances, traditional machine learning methods have become more popular in machine translation because they are better at capturing syntactic and semantic information. Researchers have employed a variety of techniques, including modeling Chinese sentence structure and word prediction using traditional machine learning methods such as HMM [8] and CRF, as well as classifying and ranking candidate translations using methods such as SVM [9]. Although traditional machine learning methods usually require domain expertise and manual feature engineering, they show good results on specific tasks and datasets. However, both SMT and traditional machine learning methods face the challenge of difficulty in handling long-distance dependencies and translation ambiguities.

In deep learning, Seq2Seq models have revolutionized machine translation. They encode source sentences into fixed-length representations and decode them into target sentences, and utilize recurrent neural networks (RNN models [10], LSTM models [11]) to solve these challenges effectively. And the attention mechanism introduced by Bahdanau and Vaswani [12] further improved the translation quality. Despite the success, it also poses a challenge in terms of computational resources, as training large neural networks requires a lot of GPU resources and memory. Meanwhile the problems of over-translation, under-translation and unnatural language generation still exist.

In recent years, researchers have made significant progress in improving Seq2Seq models to address these challenges. Technologies such as subword tokenization [13], pre-trained language models, and reinforcement learning-based fine-tuning [14], among others, have alleviated some of the limitations. These methods have improved translation quality, particularly for low-resource languages, while reducing computational demands.

However, the issue of resolving translation ambiguity still presents a significant research gap. While some efforts have explored context-aware models and semantic parsing techniques (Huang et al. [15], Zhang et al. [16]), more work is needed to provide robust solutions for translation disambiguation and effective utilization of computational resources.

In conclusion, the field of machine translation has evolved from statistical-based methods to traditional machine learning approaches, and finally to deep learning-based Seq2Seq models. While Seq2Seq models have excelled in improving translation quality, challenges related to computational resource efficiency and translation ambiguity still persist [17]. This paper aims to draw insights from this foundational research and propose a new Seq2Seq model tailored to address these specific issues.

3 N-SEQ2SEQ MODEL DESIGN

3.1 Basic Structure of Seq2Seq Model

3.1.1 Encoder and Decoder

The Seq2Seq (Sequence-to-Sequence) model is a deep learning framework for sequence-to-sequence learning, initially widely used in machine translation tasks. Its basic structure consists of an encoder (Encoder) and a decoder (Decoder).

The encoder converts each element in the input sequence into a hidden state vector, representing the entire sequence information through the composition of context vectors.

The decoder decodes the context vector and the <SOS> token to predict the next output element. It repeatedly takes the next output element and the context vector as input to make predictions, and the sequence generation ends when the generated prediction is <EOS>, completing the entire translation process.

Both the encoder and decoder use the same underlying network, typically a recurrent neural network such as the RNN model or LSTM model [18]. The Seq2Seq model is well-suited for machine translation and sequence-to-sequence tasks. Its corresponding structure is shown in Fig. 1.

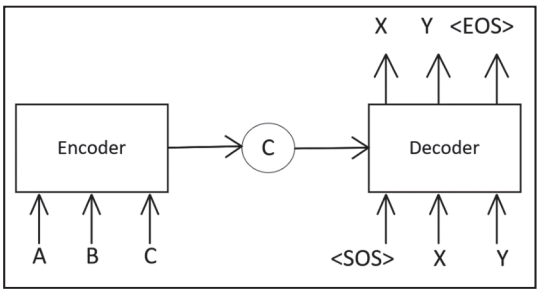

Figure 1 Encoder-decoder structure

3.1.2 Attention Mechanism

The attention mechanism is used to handle the similarity between source language texts and can allocate most computational resources to more important areas, improving the efficiency and accuracy of text processing, especially in cases of limited computing capacity. The attention mechanism allows the model to assign different weights when processing different parts of the input sequence, allowing it to focus more specifically on information that is useful for the task at hand. The basic idea is that the model relies not only on the current input but also on other parts of the input sequence when processing each time step or spatial location. This dependency is realized by calculating weights that indicate how much attention the model pays to different parts.

Self-attention mechanism [19] focuses on the information within the input sequence by learning dependencies between different positions within the same sequence. In self-attention, the sequence itself is attended to, treating each element in the sequence as a "query" matrix, a "key" matrix, and a "value" matrix. Similarities between them are calculated to construct an attention distribution. In

the self-attention mechanism, the "query" matrix, "key" matrix, and "value" matrix are all the same. The principle diagram is shown in Fig. 2.

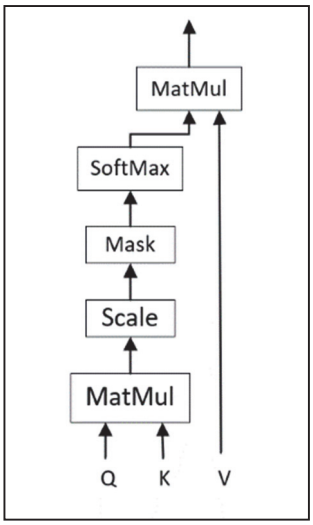

Figure 2 Self-attention mechanism diagram

3.2 Improvement Methods

3.2.1 Text Candidate Search Box

When predicting text, it is necessary to select one word or character from all the text as the current prediction result, which can slow down the prediction speed, consume memory, and greatly increase the error rate.

Therefore, this paper proposes a text candidate search box for predicting text in advance and reducing memory consumption and speeding up prediction. The text candidate search box is located in the decoder, and there is a candidate word set in the text candidate search box. The candidate word set is determined based on the previous decoder output words at the current time, and the top 10 are taken as the current prediction set. Then, each word in the candidate word set is calculated for similarity with the encoder output vector to obtain the current decoder output word. The candidate search box is connected in the decoder as shown in Fig. 3.

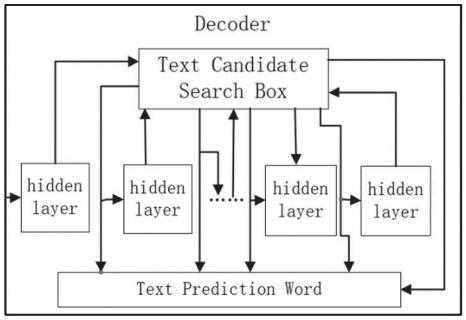

Figure 3 Diagram of candidate search box

3.2.2 Word Embedding Encoding

Initially, word vectors were represented using one-hot encoding, but when sentences are long, the dimensionality becomes extremely large, resulting in sparse encoding. This increases computational costs, and the one-hot encoding represents words as independent of each other. Due to the limitations of one-hot encoding, Word2Vec [20] became popular for training word vectors. However, it is a static word vector, meaning that once training is completed, the vector for each word remains fixed and cannot be updated with changing contexts. To address the drawbacks of Word2Vec encoding, dynamic word vectors were introduced, with common methods including the ELMo model [21], XLNet model [22], and BERT model. Among them, the BERT model is currently the best-performing dynamic word vector model. However, BERT's Chinese pre-trained model constructs word vectors at the character level, while in Chinese, most words appear as whole words [23].

Therefore, this paper uses the open-source WoBERT model to construct word vectors. The WoBERT model uses the Jieba word segmentation tool, removes redundant parts of the Bert model's built-in word list, such as Chinese words with "##," and then adds an additional 20,000 Chinese words (the most frequent 20,000 words from Jieba's built-in word list). The final size of WoBERT's vocab.txt is 33,586.

3.2.3 Relative Position Encoding

Using positional encoding can effectively represent the distance relationships between characters and words in the text. The initial use of positional encoding was in the Transformer model released in 2018. This positional encoding is essentially an absolute positional encoding, which can reflect the relative distance between characters and words but cannot indicate their chronological order [24].

This paper uses relative position encoding for representation. Relative position encoding is introduced as a set of trainable embedding encodings within the self-attention mechanism. When calculating attention, the relative distance between the current position and the position being attended to is added to the computation. This embedding encoding represents the distance between the i-th word and the j-th word. For example, a sentence of length 3 would require encoding 5 embedding vectors, as shown in Tab. 1.

Table 1 Relative position relationship index

Index	explanation
0	Relative Embedding between Position i and Position ($i-2$)
1	Relative embedding between position i and position ($i-1$)
2	Relative embedding between position i and position i
3	Relative embedding between position i and position ($i+1$)
4	Relative embedding between position i and position ($i+2$)

3.2.4 Multi-Head Attention Mechanism

The multi-head attention mechanism is an improvement over the self-attention mechanism, utilizing multiple attention mechanisms in parallel to represent the input sequence in different ways. In the multi-head attention mechanism, the input sequence is transformed into three matrices, K, Q, and V, which are then passed to multiple parallel attention heads, each of which computes a set of attention weights [25]. These weights are multiplied with the V matrix to generate a set of different representations. These representations are concatenated and transformed through a linear transformation to produce the final output of the multi-head attention. In the multi-head attention mechanism, this paper adjusts the learning of important and non-important information through a hyperparameter. This hyperparameter adjustment reduces the model parameters and enables rapid capture of crucial information. The original multi-head attention diagram is shown in Fig. 4.

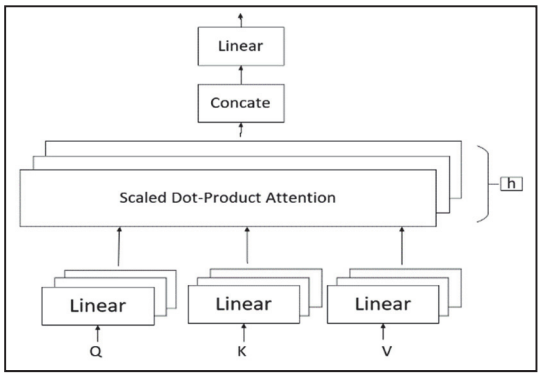

Figure 4 Multi-head attention diagram

The multi-head attention module is placed between the encoder's output and the decoder's input, serving as a pruning transformation layer [26]. Through the attention mechanism, non-important information is pruned from the output to reduce the decoder's computational resources and improve prediction accuracy.

4 EXPERIMENTAL VERIFICATION

4.1 Dataset and Experimental Environment

The training data used in this experiment includes multiple datasets, namely CMN, translation 2019zh-en, and WMT22zh-en. These datasets cover various domains such as politics, economics, technology, culture, and history, making them suitable for training and evaluating machine translation models. The WMT22zh-en dataset, provided by the WMT organization, includes a broader range of topics to support machine translation applications in various domains. The obtained data set is first preprocessed, and in machine translation, the preprocessing part of the data is crucial because it involves converting the raw text data into a format that the model can understand and process. First, the text is tokenized, i.e., the text is divided into lexemes, which can be single words or multiple characters, by means of word splitting. Then, the lexemes are de-hyphenated, discarding words or characters with a low number of occurrences. Next, vocabularies are constructed for the source and target languages, as well as for the requirements of The WoBERT model. Next, a padding operation is performed to ensure that all sentences are of the same length. Subsequently, the lexical characters are converted into a sequence format that can be processed by the model, where each token is mapped to its index in the vocabulary. Finally, the data is divided into batches and may be disrupted to increase the model's generalization capabilities. Through these preprocessing steps, the raw textual data is converted into a sequence of digits suitable for model training in preparation for subsequent training. The dataset sizes are shown in Tab. 2.

Table 2 Data size for three datasets

Dataset Name	Data Size (Unit: Ten Thousand Pairs)
CMN	3
translation2019zh-en	520
WMT22zh-en	268

After obtaining the training data, it is crucial to train the model using an appropriate hardware environment [27]. The experimental environment for this paper is shown in Tab. 3.

Table 3 Model training environment configuration

Experimental Environment	Environment Selection
Programming Language	Python 3.8
Framework	Pytorch 1.10.0
Operating System	Ubuntu 20.04
GPU	RTX 3090 (24 GB)
Cuda	11.3
Memory	43 GB

4.2 Experimental Analysis

Traditional Model: Uses LSTM network as the backbone (Seq2Seq).

Text Attention Model: Uses LSTM network as the backbone and introduces multi-head attention on the encoder's output (Seq2Seq&Attention).

Text Relative Position Encoding Model: Uses LSTM network as the backbone and introduces relative position encoding (Seq2Seq&Position).

Text Dynamic Word Embedding Model: Uses LSTM network as the backbone and incorporates WoBert model for word embeddings (Seq2Seq&WoBert).

Text Candidate Box Model: Uses LSTM network as the backbone and introduces a text candidate search box in the decoder (Seq2Seq&CandidateBox).

In the following, experiments will be conducted on these five network structures. Both ablation experiments and comparative experiments will be performed to determine if the introduced modules can improve the model's accuracy.

4.2.1 Ablation Experiment

The purpose of ablation experiments is to understand and evaluate the effectiveness of the model's network structure by dissecting the principles and components inside the model.

Experiments will be conducted based on the five network structures, and the training set's loss will be plotted as shown in Fig. 5.

Based on the loss plots for each model, the improved models in the chart are compared to the traditional model. Except for the Seq2Seq&CandidateBox model, the other models perform better than the traditional model.

Under the same number of training epochs and hyperparameters, considering that the model weights are initialized to their initial values and not yet optimized during training, the Seq2Seq&Attention, Seq2Seq&WoBert, and Seq2Seq&Position models have higher losses than Seq2Seq initially. However, as the number of epochs increases, their losses gradually decrease and eventually become lower than the traditional model's loss. Therefore, the initial higher losses in the early epochs can be considered negligible factors.

The role of Seq2Seq&CandidateBox is to accelerate the inference speed during prediction. Therefore, its loss is not significantly different from the traditional model, which is acceptable. Furthermore, because the selection of candidate words in the candidate box is related to word embeddings, combining it with text dynamic word embeddings yields better results. Additionally, as shown in the graph, with the increase in epochs, the loss of the Text Candidate Box model is slightly lower than that of the traditional model. Hence, the improvement of this model is effective.

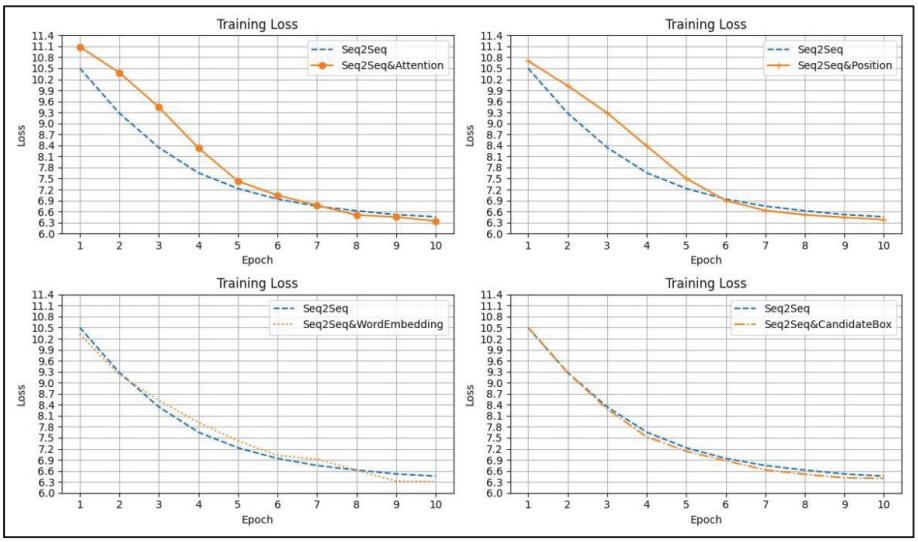

Figure 5 Loss graphs for each model in ablatio experiment

4.2.2 Comparative Experiment

Comparative experiments aim to determine which model performs better by comparing the performance of different models, with the goal of selecting the best model. Two new models are created by combining existing models. Model one: Seq2Seq + WoBert + CandidateBox + Attention; Model two: Seq2Seq + WoBert + CandidateBox + Position. These two models are compared with the traditional model, and the training loss is shown in Fig. 6.

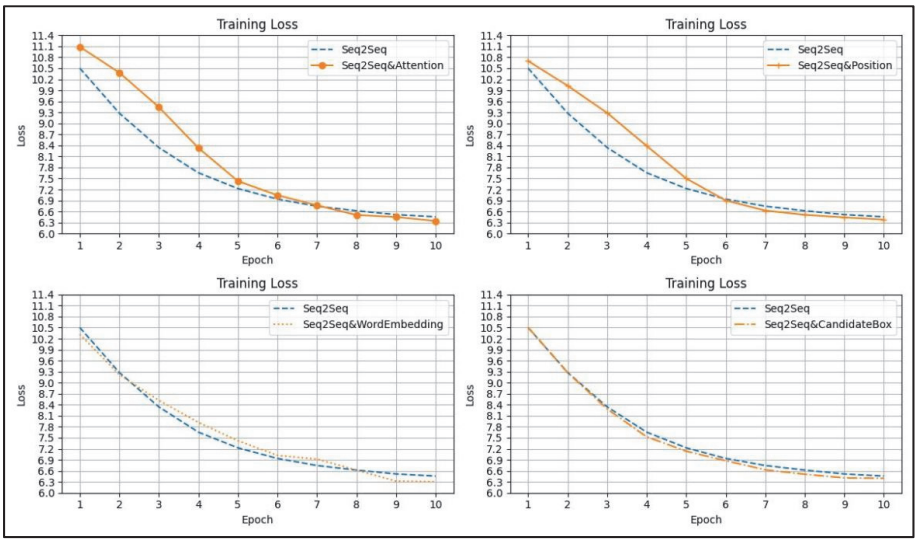

Figure 6 Loss graphs for each model in ablation experiment

Based on the loss plots for each model, the improved models in the chart are compared to the traditional model. Except for the Seq2Seq&CandidateBox model, the other models perform better than the traditional model.

Under the same number of training epochs and hyperparameters, considering that the model weights are initialized to their initial values and not yet optimized during training, the Seq2Seq&Attention, Seq2Seq&WoBert, and Seq2Seq&Position models have higher losses than Seq2Seq initially. However, as the number of epochs increases, their losses gradually decrease and eventually become lower than the traditional model's loss. Therefore, the initial higher losses in the early epochs can be considered negligible factors.

The role of Seq2Seq&CandidateBox is to accelerate the inference speed during prediction. Therefore, its loss is not significantly different from the traditional model, which is acceptable. Furthermore, because the selection of candidate words in the candidate box is related to word embeddings, combining it with text dynamic word embeddings yields better results. Additionally, as shown in the graph, with the increase in epochs, the loss of the Text Candidate Box model is slightly lower than that of the traditional model. Hence, the improvement of this model is effective.

4.2.3 Comparative Experiment

Comparative experiments aim to determine which model performs better by comparing the performance of different models, with the goal of selecting the best model. Two new models are created by combining existing models. Model one: Seq2Seq + WoBert + CandidateBox + Attention; Model two: Seq2Seq + WoBert + CandidateBox + Position. These two models are compared with the traditional model, and the training loss is shown in Fig. 7.

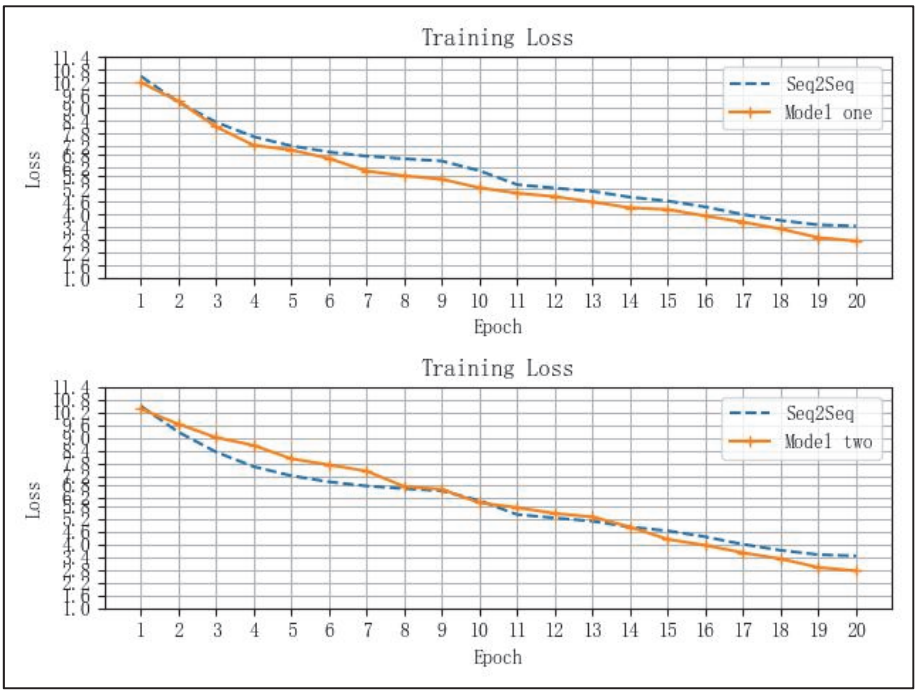

Figure 7 Loss comparison for each model in comparative experiment

From model one, it is evident that the loss is consistently lower than the traditional model, and as the number of epochs increases, the loss continues to decrease. This suggests that model one shows improvement in machine translation. Model two, on the other hand, initially has a higher loss than the traditional model in the first few training epochs. However, as the number of epochs increases, the loss gradually decreases and becomes lower than that of the traditional model. This indicates that model two also demonstrates improvement in machine translation. Moreover, around the 20th epoch, both model one and model two have lower losses than the traditional model, and their losses do not stabilize, unlike the traditional model's loss, which exhibits a flat trend. In other words, both model one and model two outperform the traditional model.

From the analysis of the models, it is evident that the role of the Attention module is primarily to filter out useless information, significantly reducing the probability of prediction errors. The Position module primarily serves to capture the relationships between words, noting the connections between the current word and the surrounding words. The WoBert module's purpose is to ensure that the segmented content aligns with the original sentence during tokenization. The main function of the CandidateBox module is to predict output results in advance and select the group with the highest probability as the final result. Introducing these modules into the original Seq2Seq model allows for faster problem-solving and consistently produces better results compared to the original Seq2Seq model.

4.3 Comparative Experiments with Transformer

In the above experiments, the final model structure was determined. The network structure modules include Seq2Seq + WoBert + Attention + Position + CandidateBox. In this context, this paper introduces the Transformer model to compare with the N-Seq2Seq model. BLEU score is used as the evaluation metric to assess the quality of translation results, calculated as shown in Eq. (1):

$$BLUE = BP \times e^{\sum_{n=1}^{N} W_n \log(P_n)} \tag{1}$$

The training parameters for comparing the N-Seq2Seq model and the Transformer model are shown in Tab. 4.

Table 4 Partial training parameters and methods for the n-seq2seq model

Relevant Parameter Settings	Numeric or Method
Training Epochs	50 epochs
Learning Rate	0.002
Number of Hidden Layers	768
Number of LSTM Layers	2
Model Training Mechanism	Tearcher forcing
Model Training Approach	Early Stopping

By training on a GPU, the initial configuration of the training environment and its related parameters as shown in Tab. (4) were set, and hyperparameter tuning was performed by adjusting various parameters to obtain a well-trained model (N-Seq2Seq).

Based on the model's training, the losses and accuracies for each epoch on the validation dataset were recorded and visualized as shown in Fig. 8.

Its BLUE evaluation scores are shown in Tab. 5.

Table 5 BLUE scores

Model	BLUE
Seq2Seq	27.32
Transformer	29.72
N-Seq2Seq	31.25

The average data inference time is presented in Tab. 6.

Table 6 Inference average time

Model	Time
Seq2Seq	0.579s
Transformer	0.368s
N-Seq2Seq	0.235s

From the loss graph, it is evident that the N-Seq2Seq model stopped training after 45 epochs, while the traditional model stopped after 47 epochs, and the Transformer model

after 39 epochs. This indicates that the N-Seq2Seq model outperforms the traditional model and shows some differences compared to the Transformer model. According to the BLUE scores calculated from Tab. 5, it is evident that the N-Seq2Seq model achieves significantly higher BLUE scores compared to both the Transformer and Seq2Seq models. As indicated by the average inference times in Tab. 6 when performing inference on the same amount of data, the N-Seq2Seq model exhibits the fastest inference speed, followed by the Transformer model, and Seq2Seq model is the slowest.

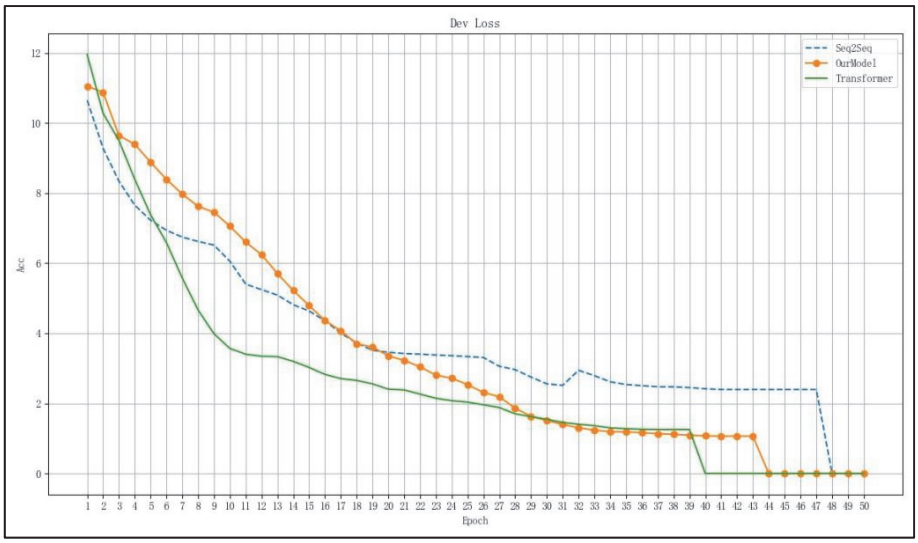

Figure 8 Loss graph

Based on the aforementioned graphs and tables, it can be concluded that the N-Seq2Seq model successfully addresses the initial problem stated in this paper. It achieves faster inference speed, accurate and efficient results, along with the advantage of being lightweight. Furthermore, it outperforms the existing Transformer model. Therefore, N-Seq2Seq can be employed as a machine translation model.

5 CONCLUSION

In this work, we introduced a novel machine translation model, N-Seq2Seq. This model employs two attention mechanisms, selectively focusing on both global and crucial information, achieved through hyperparameter tuning while filtering out non-essential details [28]. By utilizing a word-based tokenization approach and relative positional encoding, it enhances text segmentation accuracy and reinforces word associations. The introduction of text candidate frames speeds up the model's inference by making advance predictions. We conducted English-to-Chinese translation experiments on three datasets and compared the performance with Seq2Seq and Transformer models [29]. Experiments exhibit significant gains over both classic Seq2Seq and Transformer networks, displaying up to 4 BLEU and 2 BLEU score improvements. Furthermore, this experiment validated that the proposed model speeds up inference time, with efficient and accurate real-time translation on low-resource environments. When deploying the N-Seq2Seq model on mobile devices, it does not impact device performance. Additionally, when expanding to other languages with Chinese as the baseline, the model can rapidly adapt without compromising accuracy.

6 REFERENCES

[1] Sennrich, R., Haddow, B., & Birch A. (2015). Neural Machine Translation of Rare Words with Subword Units.Sennrich2015NeuralMT. *The Association for Computational Linguistics*, *1705*(5), 1715-1725.

[2] Vaswani, A., Shazeer, N., et al. (2017). Attention Is All You Need. *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 6000-6010.

[3] Devlin, J., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Association for Computational Linguistics*, 4171-4186.

[4] Ahmed, Z., Vidgen, B., & Hale, S. A. (2022). Tackling racial bias in automated online hate detection: Towards fair and accurate detection of hateful users with geometric deep learning. *EPJ Data Science, 11*(1), 8. https://doi.org/10.1140/epjds/s13688-022-00319-9

[5] Sutskever, I., Vinyals, O., & Le, Q. V. (2014). Sequence to sequence learning with neural networks. *Advances in neural information processing systems, 27*.

[6] Rudolph, J., Tan, S., & Tan, S. (2023). War of the chatbots: Bard, Bing Chat, ChatGPT, Ernie and beyond. The new AI gold rush and its impact on higher education. *Journal of Applied Learning and Teaching, 6*(1). https://doi.org/10.37074/jalt.2023.6.1.23

[7] Eidelman, V., Dyer, C., & Resnik, P. (2010). The University of Maryland Statistical Machine Translation System for the Fifth Workshop on Machine Translation. *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, 72-76.

[8] Zhao, J., Basole S., & Stamp M. (2021). Malware classification with GMM-HMM models. https://doi.org/10.5220/0010409907530762

[9] Koehn, P. (2004). Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Springer Berlin Heidelberg*, 115-124. https://doi.org/10.1007/978-3-540-30194-3_13

[10] Zaremba, W., Sutskever, I., & Vinyals, O. (2014).Recurrent neural network regularization. https://doi.org/10.48550/arXiv.1409.2329

[11] Hochreiter, S. & Schmidhuber, J. (1997).Long Short-Term Memory. *Neural Computation*, *9*(8), 1735-1780, https://doi.org/10.1162/neco.1997.9.8.1735

[12] Shaw, P., Uszkoreit, J., & Vaswani, A. (2018). Self-attention with relative position representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2, 464-468. https://doi.org/10.18653/v1/N18-2074

[13] Sennrich, R., Haddow, B., & Birch, A. (2015). Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 1*, 1715-1725. https://doi.org/10.18653/v1/P16-1162

[14] Ranzato, M. A., Chopra, S., & Auli, M., (2015). Sequence level training with recurrent neural networks.

[15] Currey, A. & Heafield, K. (2019). Incorporating Source Syntax into Transformer-Based Neural Machine Translation. *Proceedings of the Fourth Conference on Machine Translation, 1*, 24-33. https://doi.org/10.18653/v1/W19-5203

[16] Sugiyama, A. & Yoshinaga, N. (2021). Context-aware Decoder for Neural Machine Translation using a Target-side Document-Level Language Model. *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies,* 5781-5791. https://doi.org/10.18653/v1/2021.naacl-main.461

[17] Ray, P. P. (2023). ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope. *Internet of Things and Cyber-Physical Systems*, *3*, 121-154. https://doi.org/10.1016/j.iotcps.2023.04.003

[18] Wang, Z., Su, X., & Ding, Z. (2020).Long-term traffic prediction based on lstm encoder-decoder architecture. *IEEE Transactions on Intelligent Transportation Systems, 22*(10). 6561-6571. https://doi.org/10.1109/TITS.2020.2995546

[19] Lovisotto, G., Finnie, N., & Munoz, M. (2022). Give me your attention: Dot-product attention considered harmful for adversarial patch robustness, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 15234-15243. https://doi.org/10.1109/CVPR52688.2022.01480

[20] Mikolov, T., Chen, K., & Corrado, G. (2013). Efficient estimation of word representations in vector space.

[21] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark C., Lee, K., & Zettlemoyer, L. (2018). Self-Attention with Relative Position Representations. *Association for Computational Linguistics,* 2227-2237. https://doi.org/10.18653/v1/N18-1202

[22] Yang, Z., Dai, Z., & Yang, Y. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems,* 32.

[23] Cui, Y., Che, W., & Liu, T. (2021). Pre-training with whole word masking for chinesebert. *IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29*, 3504-3514. https://doi.org/10.1109/TASLP.2021.3124365

[24] Dai, Z., Yang, Z., & Yang, Y. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2978-2988. https://doi.org/10.18653/v1/P19-1285

[25] Sukhbaatar, S., Grave, E., & Bojanowski, P. (2019). Adaptive attention span in transformers. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 331-335. https://doi.org/10.18653/v1/P19-1032

[26] Bie, A., Venkitesh, B., & Monteiro, J. (2019). A simplified fully quantized transformer for end-to-end speech recognition.

[27] Yousefpour, A., Shilov, I., & Sablayrolles, A. (2021). Opacus: User-friendly differential privacy library in PyTorch.

[28] Prabha, M. I. & Srikanth, G. U. (2019).Survey of sentiment analysis using deep learning techniques. *1st international conference on innovations in information and communication technology (ICIICT). IEEE,* 1-9. https://doi.org/10.1109/ICIICT1.2019.8741438

[29] Rohde, T., Wu, X., & Liu, Y. (2021). Hierarchical learning for generation with long source sequences.

Contact information:

Yuan-shuai LAN
(Corresponding author)
School of Electronic Information Engineering, Greely university of China,
Chengdu, Sichuan 123, Section 2, Chengjian Avenue,
Eastern New Area, Jianyang City, Chengdu City, Sichuan Province, China
E-mail: 448916030@qq.com

Chuan LI
Zhongke Ruihai Dalian Intelligent Technology Research Institute Co., Ltd,
Ganjingziqu Qixian-lingjiedao Qixiandonglu, Liaoningsheng,116085,China
E-mail: m21z50c71@163.com

Xueqin MENG
School of Electronic Information Engineering, Greely university of China,
Chengdu, Sichuan 123, Section 2, Chengjian Avenue,
Eastern New Area, Jianyang City, Chengdu City, Sichuan Province, China
E-mail: 3642458430@qq.com

Tao ZHENG
School of Electronic Information Engineering, Greely university of China,
Chengdu, Sichuan 123, Section 2, Chengjian Avenue,
Eastern New Area, Jianyang City, Chengdu City, Sichuan Province, China
E-mail: 307647515@qq.com

Mincong TANG
ICIR (International Center for Informatics Research)
Room 702, 4th floor Siyuan East Building
Beijing Jiaotong University 100044 Beijing, P. R. China
E-mail: mincong@bjtu.edu.cn