International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 7171-7183

https://doi.org/10.70517/ijhsa463616

A Study on the Improvement of Cross-domain Generalization Ability of Neural Machine Translation Based on Self-supervised Learning

Jing Li[1,*]
[1] Doctoral Students, Tian Jin Foreign Studies University, Tianjin, 300204, China
Corresponding authors: (e-mail: ednaleelee@163.com).

Abstract Aiming at the problem of insufficient generalization ability of Neural Machine Translation (NMT) in cross-domain scenarios, this study proposes an LSTM-RNNs-attention model that integrates self-supervised multimodal features with an improved attention mechanism. Through multimodal self-supervised learning and text preprocessing optimization, the model constructs a graphic-text consistency classification and keyword annotation algorithm from image-text semantic correlation, which combined with the LSTM-CRF sequential word segmentation technique significantly improves the accuracy of the source language semantic representation. The experimental results show that the model F1 value reaches 99.13% when the character vector dimension is 125, and the performance is optimal when the Dropout ratio is 30% in the Chinese word segmentation task. For input noise robustness, the UNK-Tag strategy in the random word dropout mechanism has a BLEU value of 47.63 at a sampling probability of 0.15, which is 3.81% higher than the baseline. In the multilingual translation task, the BLEU scores of LSTM-RNNs-attention model on English-Chinese (Eng-Ch), Japanese-Chinese (Jap-Ch), and German-Chinese (Ger-Ch) are 45.82, 42.32, and 32.91, respectively, compared with the mainstream baseline model BERT-fused NMT, Multilingual NMT by an average of 2.6-18.0 points, and the convergence time is shortened to 6.58s (Eng-Ch), which is significantly better than the efficiency of Transformer's 16.13s and RNN-NMT's 11.79s.Manual evaluation further validates the model's semantic coherence advantage, with the Eng-Ch task scoring 9.66 points (out of 10). The study effectively solves the problem of semantic bias and long-distance dependency in cross-domain translation through self-supervised multimodal feature fusion, dynamic attention weight allocation and word segmentation optimization.

Index Terms Neural Machine Translation, Self-supervised Learning, Cross-domain Generalization, Multimodal Feature Fusion, Attention Mechanism

I. Introduction

Machine translation is an important research direction in the field of natural language processing. In recent years, with the rapid development of deep learning technology, especially the proposal of Transformer model, the ability of machine translation has been significantly enhanced [1]. Transformer-based neural machine translation system has become a paradigm for machine translation. However, the training of current neural machine translation models relies heavily on large-scale, high-quality bilingual parallel data, and the translation quality is often difficult to achieve the expected results for specific domains that lack sufficient parallel data [2]-[4]. However, domain-specific parallel data are usually scarce or non-existent, which makes improving the cross-domain generalization ability of neural machine translation an important research topic [5], [6].

The core idea of cross-domain generalization capability is to make full use of existing resources to help improve the performance of low-resource domain models [7]. In order to reduce the reliance on bilingual parallel data, there is a need to shift from a completely data-driven training model for translation models [8]. Compared to limited bilingual parallel data, resources such as domain-related monolingual data and bilingual dictionaries are more abundant and contain a large amount of domain knowledge [9], [10]. However, since these resources cannot be directly applied to the training of translation models, it leads to the limitation of the application of this knowledge in machine translation [11], [12]. Based on this, it is explored how to improve the domain generalization ability of the neural machine translation model, so that the model can better deal with domain terminology and complex sentence structures, and then build a domain-oriented data-knowledge dual-wheel-driven neural machine translation system [13]-[15].

In this study, we propose a neural machine translation model LSTM-RNNs-attention that integrates self-supervised graphic matching, serialized disambiguation and improved attention mechanism, aiming to improve the

model's adaptability to cross-domain data through multimodal feature enhancement and text preprocessing optimization. Specifically, two self-supervised tasks - graphic and text consistency classification algorithm and graphic and text keyword annotation algorithm - are firstly proposed to construct a multimodal feature extraction model from the semantic correlation between images and texts. By introducing negative example images with randomly screened ROI regions, the model learns to distinguish the matching relationship between positive and negative examples, optimizes the parameters by using the cross-entropy loss function, and finally provides graphic and textual joint representation for the translation model. Secondly, the semantic representation of the input text is optimized by the neural network segmentation model. As a key link in text preprocessing, Chinese word segmentation directly affects the translation model's understanding of input semantics. In this section, a sequence annotation model based on LSTM with Conditional Random Field CRF is used to realize the task of word segmentation with character-level annotation (B/M/E/S). Through the synergy of word vector representation layer, LSTM feature layer and CRF annotation layer, the model is able to capture contextual dependencies and reduce the need for manual feature engineering. Ultimately, multimodal features and attention mechanisms are integrated in the neural machine translation model. Based on the classical encoder-decoder architecture, the attention mechanism and multimodal feature fusion strategy are introduced. The encoder extracts text features through LSTM units and jointly encodes them with the generated image features. The decoder utilizes a dynamic attention weight allocation mechanism to enhance the ability to focus on key words. In addition, the source language embedding representation is optimized by combining the segmentation results to further enhance the generalization performance of cross-domain text.

II. Neural Machine Translation Method Based on Multimodal Self-supervised Learning and Sequence Modeling

II. A. Self-supervised Graphic Matching Algorithm Based on Graphic Relevance

II. A. 1) Overview of Self-supervised Graphic Matching Algorithms

In this paper, we denote the translated original text data and target data pairs as $\langle s,t\rangle$, and the set of candidate images for the corresponding sentences as $J_C(\langle s,t\rangle)=\{img_1,img_2,...,img_m\}$ where m is the number of images, and the set of candidate images is denoted as $J_C(\langle s,t\rangle)$ in which the image regions that have gone through the initial ROI filtering are denoted as $G_c(\langle s,t\rangle)=\{r_1,r_2,...,r_M\}$, and M is the total number of image regions in $G_C(\langle s,t\rangle)$ where the confidence level of all the images in $G_C(\langle s,t\rangle)$ satisfies the specified threshold after ROI processing respectively. The original translated text $\langle s,t\rangle$ and the filtered set of image regions $G_C(\langle s,t\rangle)$ are encoded in the deep learning model separately and the individual weights in the model are optimized by a predefined self-supervised task of graphic matching.

Applying the self-supervised graphic matching algorithm to multimodal machine translation, two different self-supervised tasks are proposed in terms of the relevance of pictures and texts: graphic consistency classification algorithm and graphic keyword labeling algorithm. Graphic consistency classification algorithm is a common self-supervised graphic matching algorithm, which mainly calculates the similarity between pictures and texts to judge their relevance by increasing the negative examples of pictures. Graphic keyword annotation algorithm is a common self-supervised serialized annotation task, which is to judge the similarity between the two by annotating the positive and negative examples of the text, so as to infer the positive and negative examples of the text through the picture. The main purpose of these two self-supervised graphic and text matching algorithms is to be able to obtain the model and parameters of image and text correlation, which provides the basis for the subsequent regional image feature screening. Since the ultimate goal of this study is to obtain image features in multimodal machine translation, these algorithms are interlocked and possess a strong before-and-after sequential relationship. The following is a detailed discussion of the graphic consistency classification algorithms.

II. A. 2) Graphic Consistency Classification Algorithm

Self-supervised graphic consistency classification algorithm is a very common way in the self-supervised task, this case is suitable to increase three groups of negative examples, the number of pictures here take five pictures, so the experiment by adding three groups of labeled pictures that have nothing to do with the original text as a negative example of the picture, and the pictures obtained by searching are merged and inputted to the model, the model through the text and the picture encoding to get the semantic vectors, at this time, respectively, the positive examples and the Negative examples of the picture feature vector and this paper to do cross-computation, and then input to the input side of the self-attention mechanism, after the full connectivity layer of the processing, to get a one-dimensional vector, due to each group has more than one picture, after the operation of taking the maximum value to get the maximum value of the vector of each group of pictures, and then with the positive and negative

examples to determine whether it is consistent with the last through the statistical computation to get the accuracy rate. It is worth noting that, since the laboratory here added three groups of negative example pictures, and the statistics is to predict the number of positive example pictures correctly, and finally the accuracy rate obtained from the calculation of statistics is compared with the standard baseline to draw conclusions. In this paper, this kind of task is called a graphic consistency classification algorithm.

For the original sentence corresponds to the picture obtained from the search, the increase of the second group, the third group, the fourth group of pictures is through a randomized algorithm to get a random number, the range of random numbers is the index range of the data set, according to the randomly obtained values and then get the picture, for the negative example of the example picture, it can clearly be seen that the positive example of the figure is obviously detailed relationship, the sentence corresponds exactly to the original sentence in the scene. On the other hand, the pictures in the second, third, and fourth groups are much different from the original pictures, which are negative example pictures, not graphic description relations.

The standard baseline for comparison here is: the first group of pictures corresponding to the text is predicted correctly, while the second, third and fourth groups are predicted incorrectly, and the four groups of pictures are predicted, and random pictures can be predicted correctly as a quarter of the results, i.e., the standard baseline is 0.25.

Ultimately, this paper calculates the loss function combined with positive and negative examples, compared with the standard baseline of 0.25, so as to conclude that if the correct rate is greater than 0.25, then this paper's image feature extraction in the picture corresponding to the text of the key feature extraction is effective, and vice versa, it is ineffective or produces negative results. The larger the difference between the two values, the more obvious the effect.

II. A. 3) Loss Function of Graphic Consistency Classification Algorithm

By modeling and learning the matching logic between graphic contents by comparing $\langle s,t\rangle$ with positive samples $G_c(\langle s,t\rangle)$ and negative samples $G_C(\langle s^{'},t^{'}\rangle)(s^{'}\neq s,t^{'}\neq t)$ to model and learn the matching logic between graphic contents. The loss function l_{tm} is computed by cross entropy CE.

$$l_{tm}=CE(p_{tm},y_{tm}) \tag{1}$$

$$p_{tm}=\frac{\exp(H_{tm}(\langle s,t\rangle,G_c(\langle s,t\rangle)))}{\sum_{1}^{n}exp((H_{tm}(\langle s,t\rangle,G_c(\langle s^{'},t^{'}\rangle))))} \tag{2}$$

As shown in Eqs. (1) and (2) above, n is the total number of positive and negative samples, and $H_{tm}(\langle s,t\rangle,G_c(\langle s,t\rangle))$ denotes the output value of the graphic matching, i.e., $\langle s,t\rangle$ and $G_c(\langle s,t\rangle)$ do the word results of the graphic consistency classification algorithm. And y_{tm} represents the one-hot vector with dimension 1.

II. A. 4) Pseudo-Code for Graphic Consistency Classification Algorithm

The pseudo-code of the graphic consistency classification algorithm provides the model parameter base for feature screening. The first 1-3 steps of the pseudo-code of the algorithm are to initialize the parameters for training and prepare the input data set, including the original text and 4 different groups of images. 4-8 steps perform cross-computation, full connectivity layer dimensionality compression, and maxima for the text and images in each batch of data set, thus calculating an intermediate value for each group of positive and negative example images for the text and thus comparing it to the correct value of the labeling, thus calculating the accuracy and loss values. The accuracy and loss values are then calculated and passed forward to the original parameters for subsequent training. The final step is to save the model parameters and model to provide a data base for subsequent feature extraction. The “similarity value” of the picture and text shown in the note in the code is an intermediate processing result between the picture and the text, and since the task is to judge the relationship between the positive and negative pictures and the description of this paper, it is more appropriate to use the “similarity value” to explain the intermediate results.

II. B.Word Slicing

After completing the extraction of graphical multimodal features, the refinement of the input text becomes the key to improve the translation quality. In this section, we will focus on the Chinese word-slicing task to optimize the semantic representation of the source language through sequence annotation model, which provides a high-quality input base for the subsequent neural machine translation model.

Generally speaking, the Chinese word slicing task is viewed as a sequence labeling problem, which is measured in characters. Specifically, each character in an input sentence is often labeled with one of B, M, E, or S. B and E denote the beginning and end of a multi-character word, M denotes the middle part, and S denotes a single-character word. There are many approaches to deal with the sequence labeling problem, such as maximum entropy Markov models in addition to conditional random fields. Recently, neural network models have been widely used in Chinese word segmentation tasks because neural network methods can minimize the effect of feature engineering.

Specifically, given a sequence $x = \{x_1, \cdots, x_n\}$ of length n characters, the goal of the Chinese word segmentation task is to compute its correctly labeled sequence $y^* = \{y_1^*, \cdots, y_n^*\}$:

$$y^* = \arg\max_y P(y \mid x) \tag{3}$$

The commonly used neural network word-splitting framework is shown in Figure 1 and consists of the following three modules:

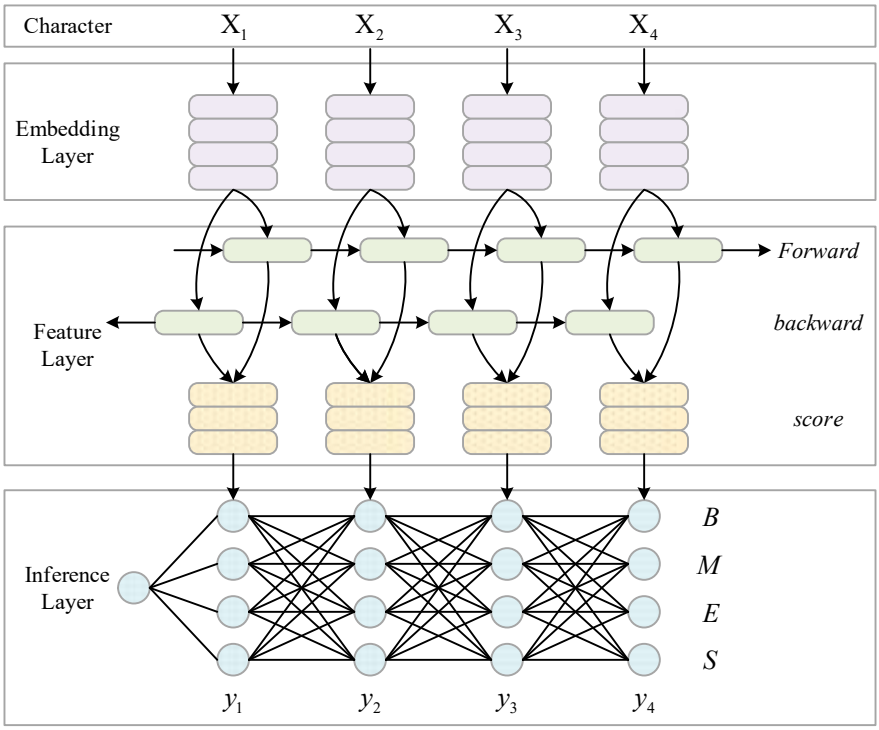

Figure 1: General Neural Network Architecture for Chinese Word Segmentation

II. B. 1) Word Vector Representation Layer

In general terms, neural network models map discrete symbolic representations to vector representations in continuous space. Formally, we find the vector $e_{x_i} \in \mathbb{R}^{d_e}$ in the matrix based on each character x_i, where d_e is a hyperparameter indicating the vector dimension of the character.

II. B. 2) Feature Layer

Extraction of features is done through feature layers, which are usually implemented utilizing the most typical neural networks. In different architectures, different models are used. Some consider long and short-term memory networks as feature layers, while others consider Transformer as a feature layer. In this paper, we use an example that considers the long and short-term memory network as a feature layer.

II. B. 3) Labeled Sequence Prediction Layer

After extracting the features, we use the conditional random field to predict the labeled sequences. The $P(y \mid x)$ in equation (3) in the conditional random field layer can be formalized as follows:

$$P(y \mid x) = \frac{\Psi(y \mid x)}{\sum_{y'} \Psi(y' \mid x)} \tag{4}$$

where $\Psi(y \mid x)$ is the potential function. Here we use a first-order linear chain conditional random field, i.e., we consider only the interaction between two consecutive labelings:

$$\Psi(y \mid x) = \prod_{i=2}^{n} \psi(x, i, y_{i-1}, y_i) \tag{5}$$

$$\psi(x, i, y' y) = \exp(s(x, i)_y + b_{y'y}) \tag{6}$$

where $b_{y'y}$ is the learnable parameter associated with the labeled pair $(y' y)$. The score function $s(x,i)$ scores each possible annotation for the i th character:

$$s(x, i) = W_s^T h_i + b_s \tag{7}$$

where h_i is the output of the feature layer at the i th position, and W_s and b_s are learnable parameters.

II. C.Neural Machine Translation

Based on the text sequence and multimodal joint features after accurate word segmentation, this section further constructs a neural machine translation model incorporating the attention mechanism. Through the dynamic feature fusion and weight assignment of encoder-decoder architecture, the generalization ability enhancement of cross-domain translation tasks is achieved.

Neural networks are not able to deal with problems such as linear indivisibility due to their simple structure, and the research was shelved. Until the back-propagation algorithm was introduced into the multilayer perceptual machine model, the feed-forward neural network was formed. With the improvement of computer parallel computing power and the application of GPUs, the problem of neural network training was solved by layer-by-layer training, which has since been applied in the fields of image and speech recognition and natural language processing. Encoder-decoder based Neural Machine Translation (NMT) was then popularized. With sufficient corpus, the translation quality of NMT is much higher than that of statistical-based machine translation.

The encoder-decoder model is the basic approach to machine translation using neural networks, also known as the seq2seq model, which solves the problem of unequal lengths of the input and output sequences. The process of translation is that the encoder encodes the input sequence, i.e., the source language, into an inter-contextual quantity, and the decoder decodes this context vector to obtain an output sequence, i.e., the target translated language.

Use x as the source language statement and Y as the target language statement, where $X = \{x_1, x_2, ..., x_n\}$, $Y = \{y_1, y_2, ..., y_n\}$. The input words are read in sequentially in the order of the input source sentence until a stopper is encountered. The encoder maps each read-in word into a fixed-length vector, and the hidden layer updates the data based on this vector, a process that can be described as shown in Equation (8):

$$h_t = f(E_x[x_t], h_{t-1}) \tag{8}$$

where $h_t \in R^d$, h_t denotes the t th processing of the input vectors performed by the hidden layer, and $f(x)$ can be a cyclic function similar to an LSTM, or it can also be viewed as a cyclic gating unit (GRU) used to update the hidden layer or other related units (e.g., memory units). The $E_x \in R^{|Y_x| \times d}$ represents the embedding matrix of the source language features, which is a query matrix that is constantly updated during the training process, where V_x is the word list, and d is the embedding matrix size.

After processing all the words in the source utterance a context vector c is generated, h_n is the summary of the input utterance with respect to the context vector c, which is used by the decoder to generate the target translated utterance. The decoder determines the probability of selecting the target word y_t based on this context vector c, the last predicted target symbols, and the state of the decoder, a process that can be described as shown in Equations (9) and (10):

$$y_t = g(E_y[y_{t-1}], s_t, c) \tag{9}$$

$$s_t = f(E_y[y_{t-1}], s_{t-1}, c) \tag{10}$$

where s_t is the hidden layer of the decoder, and since we want to compute the probability of choosing y_i as the target word, the result of the $g(x)$ function should be a number between [0, 1], of which the most commonly used $g(x)$ function in the actual training process is softmax. Both the encoder and the decoder are trained to Given the input sequence x maximizes the logarithmic probability of generating the target translation y, so the training criterion can be defined as shown in equation (11):

$$\max_{\theta} \frac{1}{K} \sum_{k=1}^{K} \log(y_k \mid x_k) \tag{11}$$

where θ is the parameter of the neural network and K is the size of the training set.

The cyclic function in the encoder-decoder model is not a simple cyclic function in the mathematical sense, the simple RNN does not capture all the features of the sequence, so LSTM-RNNs are chosen, and the LSTM unit expands the memory vector of the RNN to alleviate the problem of the RNN's long-distance dependence, and the expressions x_t, h_t and m_{t-1} are used as inputs to the LSTM, and the expressions can be summarized as shown in Eqs. (12) to (17):

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{12}$$

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{13}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{14}$$

$$g_t = \tanh(W_g x_t + U_g h_{t-1} + b_g) \tag{15}$$

$$m_t = f_t \square\ m_{t-1} + i_t \square\ g_t \tag{16}$$

$$h_t = o_t \square\ \tanh(m_t) \tag{17}$$

where i_t is the input gate, f_t is the forgetting gate, o_t is the output gate, and $\sigma(x)$ is the activation function for the corresponding multiplication of elements.

W_ρ, U_ρ and b_ρ are parameters of the neural network, where $\rho \in \{i, f, o, g\}$.

In the basic encoder-decoder structural model, each word of the input has the same degree of influence on the target translated word, but in daily life, when understanding the meaning of each sentence, each word has a different degree of influence on the understanding of the meaning of the whole utterance, for example, some intonational auxiliaries, none of them have any practical significance, and this influence feature does not appear in the basic encoder-decoder model. The article solves this problem by applying the attention mechanism to the sequence problem. Therefore, a higher level context vector is defined for each input process, and in the attention mechanism, the weights of the parameters are assigned according to the degree of influence of different words on the importance of the target translated utterance, so Eqs. (9) and (10) can be rewritten as shown in Eqs. (18) and (19):

$$y_t = g(E_y[y_{t-1}], s_t, c_t) \tag{18}$$

$$s_t = f(E_y[y_{t-1}], s_{t-1}, c_t) \tag{19}$$

where c_i is the t th h_i and s_t generation of the improvement, is defined as shown in equation (20):

$$c_i = \sum_{j=i}^{n} \alpha_{ti} h_i \tag{20}$$

where α_{ti} is the weight value and is defined as shown in equations (21) and (22):

$$\alpha_{ti} = \frac{\exp(e_{ii})}{\sum_{j=1}^{n} \exp((e_{ij})} \tag{21}$$

$$e_{ti} = s(h_i, s_{i-1}) \tag{22}$$

where $e(x)$ is a linear model reflecting the correlation between the hidden layers of the source and target languages, and $\alpha(x)$ is a hybrid function modeled by forward feedback.

III. Cross-domain Translation Performance Validation Experiment Based on Self-supervised Multimodal Features

The neural machine translation model proposed in Chapter 2, which integrates self-supervised graphic matching, serialized word segmentation and improved attention mechanism, is systematically designed from multimodal feature extraction, word segmentation optimization to dynamic weight assignment. In order to verify the generalization ability and performance advantage of the model in cross-domain translation tasks, Chapter 3 develops experimental analysis around the multilingual dataset and the baseline model, and comprehensively evaluates the robustness and efficiency of the model through the word slicing effect, the random word discarding mechanism, and the multi-task comparison experiments.

III. A. Experimental Setup

III. A. 1) Experimental Data

The data used for the word slicing experiments is the English training review corpus provided by ENG2024, the size of which is 18334 sentences and contains a total of 102,739 words. In the experiment, the data set is randomly divided into training set and test set, and the weight of the division is 8:2. When setting the parameters of the neural network model, it is necessary to set a parameter to specify the maximum length of the input sequence, and if the parameter is set too large, the training time will be increased, and the training data will be increased; and if the setting is too short, the training effect of the model will be reduced. Therefore, the maximum sentence length in the training corpus is limited to 120 by performing sentence length statistics on the corpus, so as to avoid the model spending too much training time on long sentences.

III. A. 2) Experimental Parameter Setting and Evaluation Indicators

The experimental framework based on LSTM-RNNs-attention model training is transformer, and some of the parameters are configured as follows: batchSize is set to 64; maxlen is set to 40; the number of hidden layer units of LSTM is set to 100; the dimension of English character embedding is set to 125; and the learning rate is set to 0.01; Dropout size is set to 0.3; optimization is performed using Adam's method.

Three comprehensive metrics, Precision, Recall and F-value, are used in this experiment to evaluate the performance merits of the model.

The BLEU score is used to evaluate the translation task in random word discard based studies. It measures the translation quality by comparing the similarity between the results of machine translation and one or more reference translations.

III. B. Word Slicing Experiments and Analysis

The experimental results are carried out in three main aspects: (1) using character vectors of different dimensions in the LSTM-RNNs-attention model to obtain the character vector dimensions that are most suitable for the model; (2) using different Dropout values in the LSTM-RNNs-attention model to make the model reach the optimal state; (3) comparing the different neural network models on the English word slicing task, and evaluate the effectiveness of the LSTM-RNNs-attention model based on LSTM proposed in this paper in the English word slicing task.

III. B. 1) Impact of Character Vector Dimension on Model Performance

The dimensionality of the character vectors is crucial for the accuracy of English word slicing and the training speed of the model, which will have an impact on the prediction ability of the model. In the experiments, different character vector dimensions will be set to study the effect of different dimensions on the model performance. For the size of the English word slice corpus, the character vector dimensions of 25, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, and 300 are set to study the effect of character vector dimensions on the model performance. The performance metrics of character vector effect in different dimensions are shown in Fig. 2.

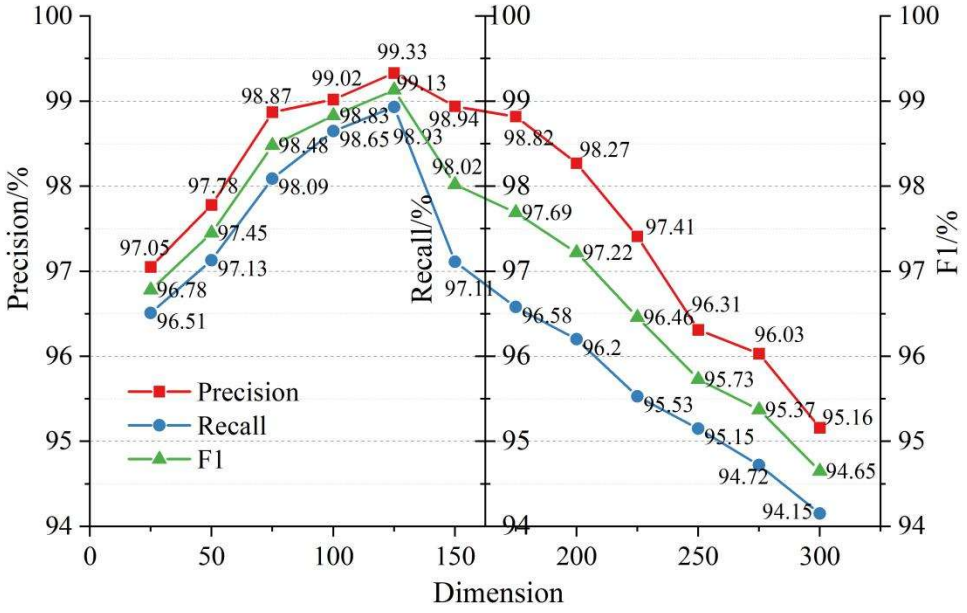

Figure 2: Performance Indicators of Character Vector Effects in Different Dimensions

As can be seen from Fig. 2, the character vector dimension of 125 works best with accuracy, recall and F1 values of 99.33%, 98.93% and 99.13% respectively. Too long vectors increase the consumption of memory space, and too short vectors do not fully express the semantics, so choosing character vectors with appropriate dimensions is very important to improve the performance of the model.

III. B. 2) Impact of Regularization on Model Performance

The idea of regularization is to avoid the problem of overfitting by simplifying the model, and Dropout is one of the effective methods. Dropout will randomly lose the activity of neurons involved in computation from the deep neural network according to a certain probability during the training process, so that they do not participate in the computation, thus avoiding the interdependence of neurons, and it is a method to reduce overfitting. This experiment investigates the effect of different Dropout's shielding probability on the model, and the performance indexes of different Dropout ratios are shown in Fig. 3.

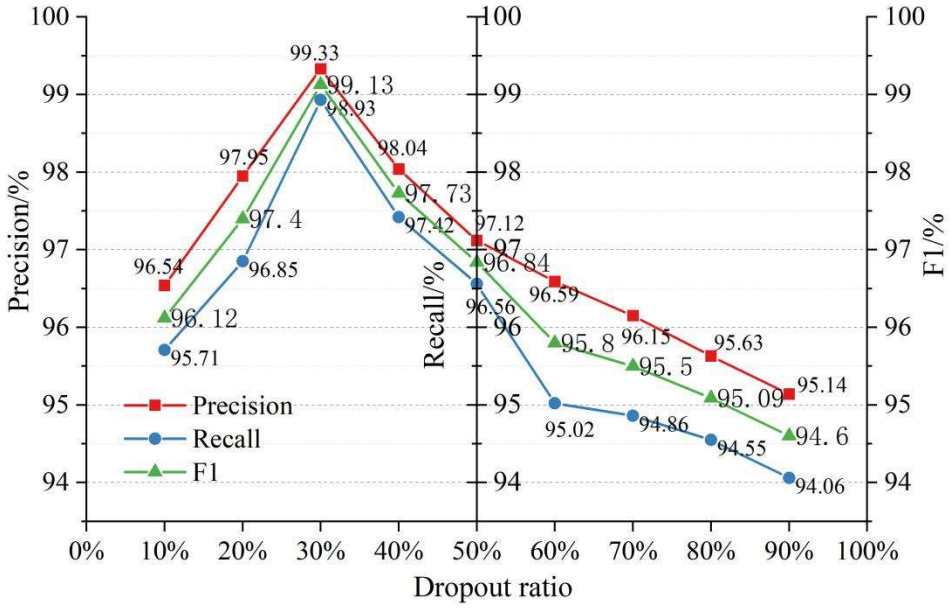

Figure 3: Performance Indicators under Different Dropout Ratios

From the experimental results in Fig. 3, it can be seen that when the proportion of Dropout value is set to 30%, the model has the best performance and the F1 value reaches 99.13%; when the proportion of Dropout value is set to 90%, the F1 value decreases significantly and reaches 94.60%. The Dropout network prevents overfitting of the deep neural network, but too high a setting tends to lead to underfitting. So certain nodes can be randomly removed to reduce the possibility of model overfitting or underfitting.

III. B. 3) Effect of Different Neural Network Models on Word Slicing

In this study, English word slice experiments were conducted using LSTM, LSTM-RNN, LSTM-RNNs and LSTM-RNNs-attention models. Firstly, the simple LSTM model was chosen as the baseline experiment, which was analyzed with the three models of LSTM-RNN, LSTM-RNNs and LSTM-RNNs-attention for the comparison experiment. In order to compare the performance of different models on the English word slicing task, the same dataset is used for the word slicing experiments in English, and the results of the four model slicing experiments are shown in Figure 4.

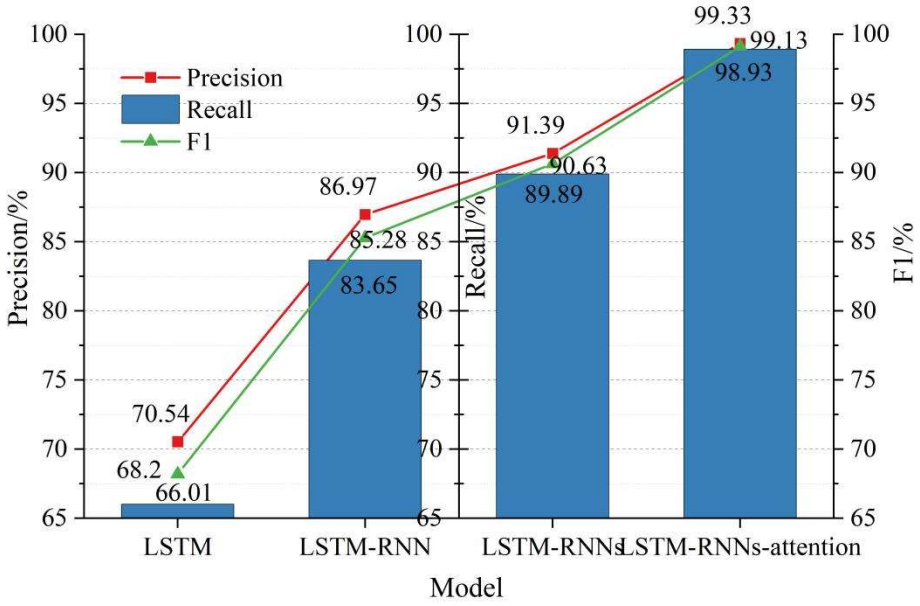

Figure 4: Results of Mongolian Word Slicing with Different Neural Network Models

The accuracy, recall and F1 values of the LSTM baseline model are 70.54%, 66.01% and 68.20%, respectively, indicating that its underlying sequence modeling capability is limited and difficult to capture complex context dependencies.The LSTM-RNN model has a significantly improved performance by introducing the RNN structure, with the three metrics reaching 86.97%, 83.65%, and 85.28%, indicating that the RNN enhances the dynamic transfer ability of sequence information. The further optimized LSTM-RNNs model with multi-layer or parameter expansion improves to 91.39% and 89.89% in accuracy and recall, respectively, with an F1 value of 90.63%, indicating that finer contextual feature extraction further optimizes the segmentation effect.

The LSTM-RNNs-attention model with the introduction of the improved attention mechanism achieves the best in the three metrics, with an accuracy of 99.33%, a recall of 98.93%, and an F1 value of 99.13%. The introduction of the attention mechanism enables the model to dynamically focus on key characters and significantly improves the ability to capture semantic associations.The F1 value is improved by 45.35 percentage points compared to the baseline LSTM and 16.24 percentage points compared to the LSTM-RNNs, which verifies the central role of the attention mechanism in reducing the misclassification and omission of scores.

III. C. Research on Neural Machine Translation Based on Random Word Discarding Mechanisms

To further verify the robustness of the model to input noise, this section introduces a random word discarding mechanism to analyze the effects of different discarding strategies on the BLEU scores of the Chinese-English translation task, and to reveal the compensating effect of self-supervised features on semantic deficiencies.

III. C. 1) Method of Discarding

The method proposed in this chapter randomly discards, rather than deletes, the input sequences of machine translation, specifically the following three types of "discarding" are used in the experiments

(1) Zero-Out of word vectors: Unlike the standard network random deactivation (Dropout) where random individual neurons are zeroed out, this method directly zeroes out the word vectors (all neurons) during the training process. The disadvantage of this method is that the contextual representation cannot be learned at the self-attentive network layer.

(2) Discarded word tag (DROP-Tag): by adding a new special symbol "<DROP>" to the word list to indicate those words that are discarded, and replacing the words that are discarded during the training process, the tag is used as a part of the word list along with other words at the same time. The label is replaced during the training process for the discarded words, and the label is updated at the same time with other words as a part of the word list.

(3) Unregistered word tag (UNK-Tag): the unregistered word "<UNK>" is used to replace the discarded words without adding new labels and parameters. This method is used on recurrent neural network RNN to force the decoder to predict the text by hidden variables.

III. C. 2) Analysis of the Probability of Dropping Samples

In order to verify the effect of the random word discarding mechanism on the translation results, we use different sampling probabilities of the training machine translation model p_s = [0.0 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40]. Figure 5 shows the variation of BLEU with different source-side dropout probabilities in the Chinese-English translation task.

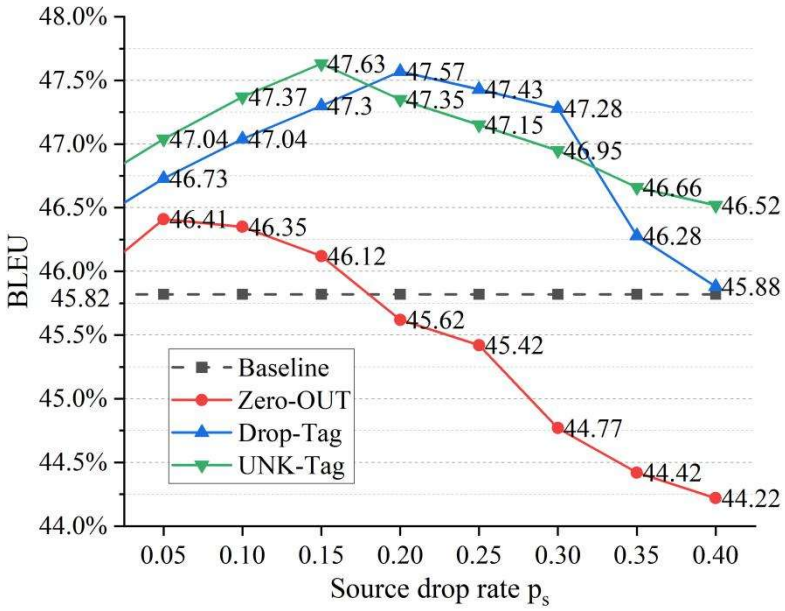

Figure 5: The BLEU Values of Different Source-end Discard Probabilities Vary

From the data trend, all three discard methods (Zero-Out, DROP-Tag, and UNK-Tag) showed a pattern of increasing and then decreasing BLEU scores when the sampling probability was varied from 0 to 0.4. Among them, the BLEU value of the Baseline model is fixed at 45.82, while the initial values of all discard methods are slightly higher than Baseline when the sampling probability is 0, suggesting that slight disturbances may help the model generalization.

Specifically, the Zero-Out method reaches a peak value of 46.41 at a sampling probability of 0.05, and then decreases continuously to 44.22 at 0.4, indicating that direct zeroing of the word vectors is only effective in a small probability range, and that corrupting the semantic information at high probability leads to a significant performance degradation.The DROP-Tag method performs better, with a BLEU value of a peak of 47.57 at a probability of 0.2, a 3.0% improvement over Baseline. The DROP-Tag method performs better, with a peak BLEU value of 47.57 at 0.2 probability, which is 3.82% higher than Baseline, and then gradually decreases to 45.88 (at p_s = 0.4).The UNK-Tag method performs the best, with a peak BLEU value of 47.63 at 0.15 probability, which is 3.81% higher than Baseline, but with the probability continuing to increase, the performance also gradually decreases to 46.52 (at p_s = 0.4).

It is worth noting that DROP-Tag and UNK-Tag perform close to each other in the medium probability (0.1-0.2)

interval, and both significantly outperform Zero-Out. at p_s = 0.15, the BLEU value of UNK-Tag is 47.63, which is slightly higher than that of DROP-Tag at 47.3, while that of Zero-Out is only 46.12. This suggests that by introducing the special tags to replace the discarded words, it can retain the contextual information more effectively and alleviate the problem of model sensitivity to missing words.

In summary, moderate random word discarding (p_s = 0.1-0.2) has a positive effect on the translation performance improvement, but it needs to be combined with a reasonable discarding strategy.UNK-Tag performs optimally at low probability (0.15), while DROP-Tag is more advantageous at high probability (0.2), and both of them significantly outperform the Zero-Out method with direct zero setting. The experimental results validate the effectiveness of the discard mechanism in enhancing the robustness of the model, while highlighting the critical impact of strategy selection on the final performance.

III. D. Multi-task Model Comparison Experiment

Based on the local optimization results of the prelude experiments, this section extends to the multilingual translation task, and comprehensively verifies the significant advantages of this paper's model in terms of cross-domain generalization ability by comparing the BLEU scores, convergence efficiencies, and manually-measured performances of the mainstream baseline models.

III. D. 1) Data Sets

The effectiveness of the self-supervised learning-based neural machine model proposed in this paper is verified in different translation tasks, including English-Chinese (Eng-Ch), Japanese-Chinese (Jap-Ch), and German-Chinese (Ger-Ch), and the data used are obtained from CWMT2020, Wikipedia Kyoto Articles/WMT2020, and OPUS9. In order to avoid the model allocating too much training time on long sentences, all sentence pairs with a length of more than 50 at the source or target end are discarded.

III. D. 2) Baseline Model

In order to comprehensively evaluate the advantages of the self-supervised multimodal model proposed in this paper in terms of cross-lingual generalization ability, six popular baseline models are selected for comparison.

(1) Transformer: a classical model based on the self-attention mechanism, which completely abandons the loop and convolutional structure and captures long-distance dependencies through the multi-head attention mechanism, is the current mainstream baseline for neural machine translation.

(2) RNN-NMT: an encoder-decoder architecture using LSTM or GRU, relying on recurrent neural networks to process sequence information and conveying context through hidden states, a representative model of early neural machine translation.

(3) ConvS2S: a sequence-to-sequence model based on convolutional neural networks, utilizing layer-by-layer convolution to extract local features and preserving sequence order through positional encoding, with parallel computing advantages.

(4) BERT-fused NMT: combining the pre-trained language model BERT with Neural Machine Translation, which utilizes the contextual representation of BERT to enhance the source language coding and improve the ability to capture complex semantics.

(5) MASS (Masked Sequence-to-Sequence): a masked pre-trained sequence generation model based on large-scale monolingual data pre-training, with excellent performance in low-resource translation tasks.

(6) Multilingual NMT: a multilingual joint training model that learns cross-language generalized features through shared parameters and is suitable for migration learning of resource-scarce language pairs.

III. D. 3) Machine Translation Quality Assessment

The results of the comparison of each baseline model in terms of machine translation quality are shown in Table 1.

Table 1: The Performance of Each Model on the BLEU Score

Model	Eng-Ch	Jap-Ch	Ger-Ch
Transformer	27.78	24.77	17.16
RNN-NMT	31.45	29.79	24.63
ConvS2S	30.37	34.91	23.28
BERT-fused NMT	43.19	36.62	26.07
MASS	40.72	31.16	29.25
Multilingual NMT	41.09	40.98	23.65
LSTM-RNNs-attention	45.82	42.32	32.91

The LSTM-RNNs-attention model achieves the highest BLEU scores of 45.82, 42.32, and 32.91 in the English-Chinese (Eng-Ch), Japanese-Chinese (Jap-Ch), and German-Chinese (Ger-Ch) tasks, which are significantly better than the baseline model. For example, in the Eng-Ch task, its score improves by 2.63 points over the 43.19 of the BERT-fused NMT and nearly 18 points over the 27.78 of the Transformer, verifying the effectiveness of the self-supervised multimodal feature and attention mechanism. Traditional models such as RNN-NMT (Eng-Ch:31.45) and ConvS2S (Jap-Ch:34.91) perform moderately well, while the pre-trained models BERT-fused NMT and MASS reach 43.19 and 40.72 in the Eng-Ch task, respectively, but are still lower than the models in this paper. Notably, Multilingual NMT performs outstandingly in the Jap-Ch task at 40.98, but the Ger-Ch score is only 23.65, indicating that multilingual co-training has limited effect on the migration of resource-scarce language pairs.

III. D. 4) Comparison of Convergence Time and Efficiency Statistics

In order to measure the impact of the translation models on the translations, the experiments also compared the training convergence time, the change of BLEU with each training stage during the training process, and the final translation example review. The convergence time and efficiency statistics of different models are shown in Table 2.

Table 2: Statistics of Convergence Time and Efficiency of Different Models

Model	Eng-Ch	Jap-Ch	Ger-Ch
Transformer	16.13s	22.30s	29.76s
RNN-NMT	11.79s	17.87s	24.12s
ConvS2S	16.59s	19.25s	22.21s
BERT-fused NMT	14.22s	15.02s	21.35s
MASS	10.06s	13.06s	16.37s
Multilingual NMT	9.15s	12.65s	15.74s
LSTM-RNNs-attention	6.58s	10.16s	12.75s

The LSTM-RNNs-attention model has the shortest convergence time among all language pairs, taking only 6.58s in Eng-Ch, which is much lower than Transformer's 16.13s and RNN-NMT's 11.79s. Its training efficiency is especially significant in the more resource-consuming Ger-Ch task, with 12.75s, which is superior to that of the Multilingual NMT's 15.74s.Pre-trained models such as MASS at 10.06s in Eng-Ch and Multilingual NMT's 15.74s in Ger-Ch are more efficient, but there is a gap between the translation quality and that of the model in this paper.ConvS2S takes 19.25s in the Jap-Ch task, but its BLEU score of 34.91 is lower than the 42.32 of this paper's model, indicating that its efficiency and performance are not balanced. Overall, this paper's model significantly improves the training efficiency while ensuring the translation quality through multimodal feature fusion and optimized attention mechanism.

III. D. 5) Manually Measured Performance

A single BLEU evaluation cannot perfectly measure the semantic fluency of a sentence. Therefore, the experiment uses the manual evaluation specification in the Machine Translation Evaluation Syllabus as a standard to evaluate the best translation results. Based on the comprehensibility in the syllabus, the sentence scores range from 0 to 10, including two decimals, and the final score is the arithmetic mean of all scores. The manual assessment performance of each model is shown in Table 3.

Table 3: The Manual Evaluation Performance of Different Models

Model	Eng-Ch	Jap-Ch	Ger-Ch
Transformer	6.66	5.57	5.91
RNN-NMT	7.48	7.11	7.18
ConvS2S	8.76	8.43	8.22
BERT-fused NMT	8.33	8.05	7.81
MASS	8.58	7.73	7.04
Multilingual NMT	9.29	9.03	9.37
LSTM-RNNs-attention	9.66	8.86	9.21

The manual scoring results show that the LSTM-RNNs-attention model has the highest score of 9.66 in the Eng-Ch task and reaches 9.21 in the Ger-Ch task, which are both significantly better than the other models. Its Eng-Ch score is 0.37 points higher than Multilingual NMT's 9.29, and 1.33 points higher than BERT-fused NMT's 8.33, indicating that its translation fluency and semantic coherence are better. Notably, Multilingual NMT has a higher BLEU score of 40.98 in the Jap-Ch task, and its manual score of 9.03 is slightly higher than that of the model in this paper, which is 8.86, indicating that the sequence generation model based on mask pre-training outperforms the model in this paper in the translation of Japanese. Traditional models such as RNN-NMT have a lower rating of 7.48 in Eng-Ch, reflecting the mechanical and contextual disconnection problem of their generated utterances. In summary, the overall lead of this paper's model in manual evaluation further validates its advantages in cross-domain generalization and semantic understanding.

IV. Conclusion

In this study, an LSTM-RNNs-attention model incorporating self-supervised multimodal features with an improved attention mechanism is proposed to address the bottleneck of generalization ability of neural machine translation in cross-domain scenarios.

In the Chinese segmentation task, when the model adopts 125-dimensional character vectors with 30% Dropout strategy, the F1 value reaches 99.13%, which is 45.35 percentage points higher than the baseline LSTM, indicating that the serialized segmentation and multimodal features co-optimization can effectively reduce the problems of misclassification and omission.

For the input noise interference, the UNK-Tag strategy in the random word discarding mechanism improves the BLEU value to 47.63 at the sampling probability of 0.15, which is 3.81% higher than the baseline of 45.82, verifying the ability of the self-supervised features to compensate for the semantic deficit.

In the multilingual translation task, the LSTM-RNNs-attention model achieves BLEU scores of 45.82, 42.32, and 32.91 on English-Chinese, Japanese-Chinese, and German-Chinese, respectively, which is an average improvement from the baseline models of 43.19 for the BERT-fused NMT and 40.98 for the Multilingual NMT, etc. 2.6-18.0 scores, indicating that multimodal self-supervised learning significantly enhances cross-domain semantic migration. Meanwhile, the model convergence time is shortened to 6.58s in the Eng-Ch task, which is more than 60% more efficient than Transformer's 16.13s and RNN-NMT's 11.79s, highlighting the technical advantages of the dynamic attention mechanism and parameter optimization. Manual evaluation further verifies the semantic coherence of the model, and the Eng-Ch task scores 9.66 points, which is a significant improvement over the 7.48 points of the traditional model RNN-NMT.

References

[1] Stahlberg, F. (2020). Neural machine translation: A review. Journal of Artificial Intelligence Research, 69, 343-418.

[2] Castilho, S., Moorkens, J., Gaspari, F., Calixto, I., Tinsley, J., & Way, A. (2017). Is neural machine translation the new state of the art?. The Prague Bulletin of Mathematical Linguistics, (108).

[3] Zheng, Z., Zhou, H., Huang, S., Li, L., Dai, X. Y., & Chen, J. (2019). Mirror-generative neural machine translation. In International Conference on Learning Representations.

[4] Firat, O., Cho, K., Sankaran, B., Vural, F. T. Y., & Bengio, Y. (2017). Multi-way, multilingual neural machine translation. Computer speech & language, 45, 236-252.

[5] Karakanta, A., Dehdari, J., & Van Genabith, J. (2018). Neural machine translation for low-resource languages without parallel corpora. Machine Translation, 32, 167-189.

[6] Wu, S., Zhang, D., Zhang, Z., Yang, N., Li, M., & Zhou, M. (2018). Dependency-to-dependency neural machine translation. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 26(11), 2132-2141.

[7] Saunders, D. (2022). Domain adaptation and multi-domain adaptation for neural machine translation: A survey. Journal of Artificial Intelligence Research, 75, 351-424.

[8] Pham, M., Crego, J. M., & Yvon, F. (2021). Revisiting multi-domain machine translation. Transactions of the Association for Computational Linguistics, 9, 17-35.

[9] Zhang, Z., Liu, S., Li, M., Zhou, M., & Chen, E. (2018, April). Joint training for neural machine translation models with monolingual data. In Proceedings of the AAAI conference on artificial intelligence (Vol. 32, No. 1).

[10] Xu, G., Ko, Y., & Seo, J. (2019). Improving neural machine translation by filtering synthetic parallel data. Entropy, 21(12), 1213.

[11] Shi, S., Wu, X., Su, R., & Huang, H. (2022). Low-resource neural machine translation: Methods and trends. ACM Transactions on Asian and Low-Resource Language Information Processing, 21(5), 1-22.

[12] Yang, Z., Chen, W., Wang, F., & Xu, B. (2019). Effectively training neural machine translation models with monolingual data. Neurocomputing, 333, 240-247.

[13] Chu, C., & Wang, R. (2020). A survey of domain adaptation for machine translation. Journal of information processing, 28, 413-426.

[14] Ala, H., & Sharma, D. M. (2020, December). AdapNMT: Neural Machine Translation with Technical Domain Adaptation for Indic Languages. In Proceedings of the 17th International Conference on Natural Language Processing (ICON): Adap-MT 2020 Shared Task (pp. 6-10).

[15] Buonocore, T. M., Crema, C., Redolfi, A., Bellazzi, R., & Parimbelli, E. (2023). Localizing in-domain adaptation of transformer-based biomedical language models. Journal of Biomedical Informatics, 144, 104431.