

Research Article

Application of Machine Learning Combined with Wireless Network in Design of Online Translation System

Jingyun Li 

Department of Foreign Languages, University of Sanya, Sanya, 572022 Hainan, China

Correspondence should be addressed to Jingyun Li; jingyunli@sanyau.edu.cn

Received 18 July 2022; Revised 22 August 2022; Accepted 2 September 2022; Published 21 November 2022

Academic Editor: Kuruva Lakshmana

Copyright © 2022 Jingyun Li. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The importance of translation services has become increasingly prominent with the acceleration of economic globalization. Compared with human translation, machine translation is cheaper and faster, and therefore more suitable for the current era. The current mainstream machine translation method is neural machine translation, which employs machine methods to train on parallel corpora and create translation models. Research into neural machine translation has yielded a wealth of information. Learning and generalization abilities of neural networks have substantially enhanced the effectiveness of neural machine translation. This work applies machine learning and wireless network technology to build an online translation system for real-time translation. First, this work proposes a multigranularity feature fusion method based on a directed acyclic graph, which uses a directed acyclic graph to fuse different granularities as input and obtain a position representation. Secondly, this paper improves the Transformer model and proposes multigranularity position encoding and multigranularity self-attention. Then, on the basis of multigranularity features as input, this work introduces dynamic word vectors to improve the word embedding module, and uses the ELMo model to obtain dynamic word vector embeddings. Finally, this work builds a multigranularity feature-dynamic word vector machine translation model with above strategy, deploys it on server. Users can upload the content to be translated and download the translated content through the wireless network and realize an online translation system based on machine learning and wireless network.

1. Introduction

Machine translation is an experimental discipline that uses electronic computers to translate between natural languages. As a subfield of computational linguistics, machine translation is an interdisciplinary field of study. Machine translation, on the other hand, is based on the three fields of linguistics, mathematics, and computer science. People need to integrate the results of researchers in different fields such as linguists, mathematicians, and computer technology experts to achieve machine translation. Without the efforts of any of them, machine translation is difficult to achieve [1, 2]. Due to the development of deep learning algorithms and the enhancement of hardware computing power, machine translation based on neural networks has developed rapidly. It simplifies many complicated operations when designing machine translation systems in the past and achieves better results than traditional methods. With the advent of personal computers

and the shift toward translation memory tools for translators, machine translation has been put into practice [3, 4]. The trend in recent years is to combine some methods of traditional statistical machine translation with neural network methods or purely neural network-based methods, especially neural network-based end-to-end machine translation. This method abandons the complicated procedures of traditional statistical machine translation and directly uses parallel corpus as input and output for end-to-end training. At the same time, it can solve issue of long-distance dependence, and it has become the mainstream research of major companies and institutions, and it will also become the main research direction in the next few decades [5].

A typical multidisciplinary research issue, machine translation encompasses a wide range of fields. There is a significant deal of theoretical value in conducting machine translation research because it can aid the growth of many different fields such as cognitive and computational

linguistics as well as artificial intelligence [6, 7]. The development of other natural language processing tasks, such as named entity recognition, sentiment analysis, and autonomous text synthesis, has also been influenced by research on machine translation. In terms of application, whether it is the general public, government enterprises, or national institutions, machine translation technology is urgently needed [8, 9]. Machine translation technology is not yet perfect, and it still faces many specific problems and difficulties, which require the unremitting efforts of relevant researchers [10].

The rule-based machine translation process can generally be divided into analysis and conversion as well as generation. Analysis refers to analysis for morphological and syntactic structure of the original text, conversion refers to the mutual conversion between the original text and the translation, and generation refers to the generation of the required translation [11]. The first generation of machine translation is word-based machine translation. In the process of translation, this machine translation converts the words of the original text into the words of the translation, and the translation effect is very poor. The second generation of machine translation is syntactic-based machine translation. In addition to lexical conversion during translation, the analysis of syntactic structure and the generation of syntactic structure are also emphasized. The translation quality of this machine translation has been slightly improved compared to word-based machine translation. The third generation of machine translation is semantic-based machine translation. When translating, the original text is first analyzed semantically to obtain the semantic content of the original text, and then this semantic content is represented by the text of the translation. It is rare to see rule-based machine translation in use right now [12, 13]. Rules are more commonly used with other machine translation approaches. It is possible to separate corpus-based machine translation approaches into instance-based machine translation methods, statistical machine translation, and deep learning-based machine translation. Instance-based translation methods do not require complex analysis of source language sentences and make full use of confirmed translation instances. However, since the instances used by this method are generally sentences. How to quickly find instances with high similarity from a large-scale instance library is always a challenge for this method. Statistical machine translation can be further divided into word-based translation models, phrase-based translation models, and syntactic tree-based translation models [14]. Due to the unique advantages of deep learning methods in feature representation and end-to-end modeling, deep learning-based methods have gradually become a research hotspot in machine translation. Machine translation with deep learning mainly uses statistical machine translation as a framework, aiming to improve key technologies in source language sentence parsing, translation conversion, and target translation generation [15].

This work applies machine learning and wireless network technology to build an online translation system for real-time translation. First, this work proposes a multigranularity feature fusion method based on a directed acyclic

graph, which uses a directed acyclic graph to fuse different granularities as input and obtain a position representation. Secondly, this paper improves the Transformer model and proposes multigranularity position encoding and multigranularity self-attention. Then, on the basis of multigranularity features as input, this work introduces dynamic word vectors to improve the word embedding module and uses the ELMo model to obtain dynamic word vector embeddings. Finally, this work builds a multigranularity feature-dynamic word vector machine translation model with above strategy, deploys it on server. Users can upload the content to be translated and download the translated content through the wireless network and realize an online translation system based on machine learning and wireless network.

2. Related Work

Reference [16] proposed an end-to-end machine translation method based on neural network, which used convolutional neural network to extract source language features and used RNN to decode source language feature vector into target language. Reference [17] implemented a fully neural network-based RNN-RNN sequence-to-sequence machine translation model. Reference [18] proposed to use LSTM to replace the CNN and RNN of the aforementioned models. LSTM controlled the transmission of information by adding a gate mechanism, which greatly alleviated the problems of gradient disappearance and gradient explosion. Reference [19] proposed to introduce attention mechanism into neural machine translation, which was first proposed by DeepMind for image classification. The attention mechanism could dynamically obtain the source language word information related to the generated words during decoding. It solves the problem of fixed-length vectors and significantly improved the translation effect. It was an important research progress in neural machine translation. After this, the attention encoder-decoder network became the state-of-the-art model for NMT. Reference [20] proposed a local attention model, which improved the global attention, which could reduce part of the computation and improve the translation quality. Reference [21] fully used CNN to implement NMT, and its translation performance was comparable to RNN-based NMT, however the training speed was 9 times faster. The improvement work of Transformer translation quality improvement was more focused on the improvement of self-attention mechanism, and the current research was mainly divided into two categories. One approach was to add additional information to optimize the calculation of attention weights, and the other approach was to improve the way the weights of the self-attention mechanism are calculated. Reference [22] introduced a new position matrix to calculate a Gaussian bias term with the intermediate state of self-attention. It was added to the original self-attention distribution calculation to make the weight distribution smoother to improve the ability to capture short-range semantic dependencies. Reference [23] replaced the masked self-attention mechanism on the decoder side with an average attention mechanism, which speeded up the decoding speed of the model while ensuring comparable model effects.

Reference [24] replaced the multihead self-attention mechanism with weighted multibranch self-attention. The weighted improvement could assign different weights to different points of multiplication attention and learn the different degrees of attention of each branch, which can improve the performance of the model. Reference [25] used a convolutional neural network to extract the source language sentence-level topic context and added the topic module to the RNN translation model and the Transformer translation model, which improved the translation quality of the model

The original Transformer model generated translations in a sequential manner. Therefore, the translation model could not achieve parallel generation of translations in the inference stage, which affected the efficiency of the model. To solve this problem, a non-autoregressive Transformer model was proposed, which had the same structure on the encoder side as the autoregressive Transformer. While on the decoder side, instead of relying on previously generated tokens, all translations are directly generated at one time step. Reference [26] proposed a non-autoregressive neural machine translation framework. Reference [27] used phrase table and word-level adversarial methods to improve the NAT model, which enhanced the target language information in the decoder input and improves the translation quality. Reference [28] generated a short sequence of discrete latent variables through autoregression, and then used this latent variable as a condition to generate all target tokens non-autoregressively, but the calculation of latent variables was too complicated. Reference [29] proposed a grammar-supervised Transformer, which first autoregressively predicted a parsing sequence of grammar blocks. All target tokens were then generated at once based on the predicted parsing, but model performance was heavily dependent on the generated parsing blocks. Therefore, making the parsing decoder generate more accurate grammar block sequences could improve the translation effect of the model. Reference [30] normalized NAT by introducing two auxiliary regularization terms, similarity and reconstruction, into the training target, which alleviated the problem of over- and missing-translations in translation models. Reference [31] proposed two methods for retrieving target sequence information in order to solve the problem of model over- and missing-translation. One was to apply reinforcement learning to non-autoregressive Transformer, and the other was to propose a new Transformer decoder FS-decoder, which fused target sequence information into the top layer of the decoder. Reference [32] also found that word-level cross-entropy loss in non-autoregressive translation could not model the order dependencies on the target side well. This resulted in a weak correlation with translation quality, so it was proposed to minimize the BoN difference between the model output and the reference sentence to train the model. The proposed BoN training objective was differentiable and could be computed efficiently, which was beneficial for NAT to capture sequence dependencies on the target side, and the BoN training objective had a good correlation with translation quality. Reference [33] added a context encoder to the structure of the traditional Transformer to encode the above

information before the current sentence. The training was divided into two steps, first is training a basic machine translation model with sentence-level parallel corpus. Then, a context encoder was added to this model, and the document-level parallel corpus was used for training. At this time, the part of the basic model previously trained with the sentence-level parallel corpus would be frozen. Their models showed significant improvements over the baselines on the NIST Chinese-to-English and IWSLT French-to-English tasks. Compared with complex models such as multiple encoders, literature [34] proposed a simple and effective document-level machine translation model. It only used one encoder to encode the context and the source sentence at the same time, and at the top layer of the encoder, only part of the source sentence was encoded and output to the decoder. The model was simpler than other multienncoder structures and achieved better improvement on English to German tasks. Reference [35] used the word embedding information learned by the sentence-level model, and took the average value of all word embedding information in the document as the document embedding information. By introducing a document label for each sentence and replacing the label with the document's embedding information when embedding, the model was made aware of the global information of the entire document. With such a small introduction of information, there was still a significant improvement on the English to German dataset.

3. Method

This work applies machine learning and wireless network technology to build an online translation system for real-time translation. First, this work proposes a multigranularity feature fusion method based on a directed acyclic graph, which uses a directed acyclic graph to fuse different granularities as input and obtain a position representation. Secondly, this paper improves the Transformer model and proposes multigranular position encoding and multigranular self-attention. Then, on the basis of multigranularity features as input, this work introduces dynamic word vectors to improve the word embedding module, and uses the ELMo model to obtain dynamic word vector embeddings. Finally, this work builds a multigranularity feature-dynamic word vector machine translation model based on the above strategy and deploys it on the server. Users can upload the content to be translated and download the translated content through the wireless network and realize an online translation system based on machine learning and wireless network.

3.1. Recurrent Neural Network and Attention. For machine translation tasks, whether the source language sentence or the target language sentence is a natural sentence, there is an order and position relationship. In general neural networks, each neuron is independent and parallel, and cannot process sequence input. RNN came into being, its specialty is processing sequence data. Due to its network characteristics, the recurrent neural network has the function of memory. The network will remember the input information at the previous moment and affect the output at the current

moment. Figure 1 is a schematic diagram of a simple recurrent neural network, which consists of an input layer, a hidden layer, and an output layer.

The machine translation model based on the above-mentioned recurrent neural network has achieved certain results in translation research with shorter lengths, but it encounters the problems of gradient disappearance and gradient explosion when dealing with longer sequences. The reason is that the hidden state transfer or back-propagation of the recurrent neural network will multiply the weight matrix. Assuming that the weight matrix can be decomposed by eigenvalues, if the absolute value of an eigenvalue is less than 1, it will approach 0 after continuous multiplication, causing the gradient to disappear. If the absolute value of the eigenvalue is greater than 1, it will surge after continuous multiplication, causing the gradient to explode.

LSTM has achieved excellent performance on machine translation tasks. LSTM introduces a gate mechanism, which effectively alleviates the problem of gradient disappearance. The input gate controls the influence of the input information at the current moment on the hidden state at the current moment. The output gate controls the influence of the hidden state at the current moment on the output information at the current moment. The forget gate controls the memory unit and chooses to forget the hidden state information of the historical moment. LSTM calculations are

$$f_t = \sigma(W_f[h_{t-1}; x_t] + b_f), \quad (1)$$

$$i_t = \sigma(W_i[h_{t-1}; x_t] + b_i), \quad (2)$$

$$o_t = \sigma(W_o[h_{t-1}; x_t] + b_o), \quad (3)$$

$$h_i = o_t * \tanh(c_t), \quad (4)$$

$$c_t = f_t * c_{t-1} + i_t * \tanh(W_s[h_{t-1}; x_t] + b_s), \quad (5)$$

where W is weight, x is input, and b is bias.

The most basic neural machine translation model is based on the encoder-decoder framework. First, the encoder encodes the source language sentence into a fixed-length semantic vector, and then the decoder uses the semantic vector to continuously generate the target word. Obviously, fixed-length semantic vectors have limited representation ability and cannot well express all the information contained in source language sentences. Especially when the input sentence is very long, the effect of the traditional neural machine translation model is often not satisfactory. In order to solve the above problems, an attention mechanism is introduced, that is, there is a one-to-one correspondence between the source language and the target language.

The introduction of attention mechanism can reduce the computational burden of processing multidimensional data input and compress the data representation by selecting a subset of the input in a structured way. At the same time, it can also grasp the focus, so that the prediction system can focus more on processing the key information in the input data that is significantly related to the current output, thereby improving the quality of the output. The attention mechanism is essentially an automatic weighting scheme.

With the introduction of the attention mechanism, each time step model will perform a weighted summation of all the hidden vectors of the encoder according to the automatically calculated weight probability and obtain a new context vector. Because the weights corresponding to the hidden layers at each time step are different, the input context received by the decoder at each time step is no longer fixed. This enables the decoder at each time step to focus on processing the most relevant information from the original module to the current output.

By introducing an attention mechanism, an alignment network from the source language sequence to the target language sequence is actually established. After being weighted by the attention mechanism, the original fixed-length and invariant semantic vector will become a dynamic semantic vector matrix. This greatly enhances the ability to represent long sentences. The attention in LSTM is demonstrated in Figure 2.

3.2. Multigranularity Feature Fusion via DAG. When the information of word granularity needs to be used, it is necessary to manually segment the text. The current word segmentation has the problem of inconsistent segmentation standards. For the language, words are the most basic information units. In machine translation systems, input sentences are generally divided based on character granularity or word granularity. In general, translation systems based on word-granularity have better performance than translation systems based on character-granularity. This also proves from the side that words or phrases contain more information than simple characters. When the word granularity division is not uniform, the word granularity with the best word segmentation effect under the specific task will be selected for division. The above two granularity division methods are single, and it is impossible for any single word segmentation granularity to completely represent the information contained in the sentence, thus causing a certain loss to the information of the input sentence. This work proposes a multigranularity feature fusion method that utilizes a directed acyclic graph (DAG) for multigranularity representation. It can be characterized by combining character granularity and multiple word granularity through a directed acyclic graph. Figure 3 is a directed acyclic graph.

When a sequence is input to a recurrent neural network, it is generally input character by character or word by word. When inputting a convolutional neural network or a Transformer model, since it is a parallel input, the corresponding position representation must be added. In the original Transformer model, word embeddings and their corresponding position codes are fused in the form of summation, and multiplication and splicing are common. This chapter uses the directed acyclic graph to fuse the multigranularity features, and the position representation can also be obtained by using the directed acyclic graph. In a directed acyclic graph, edges connect each character node, and any word is also connected by edges. Suppose a word consists of node v_i to node v_{i+k} , then the edge connecting them is $e_{i:i+k}$. Taking advantage of this feature, this chapter proposes a two-layer position representation. For the word

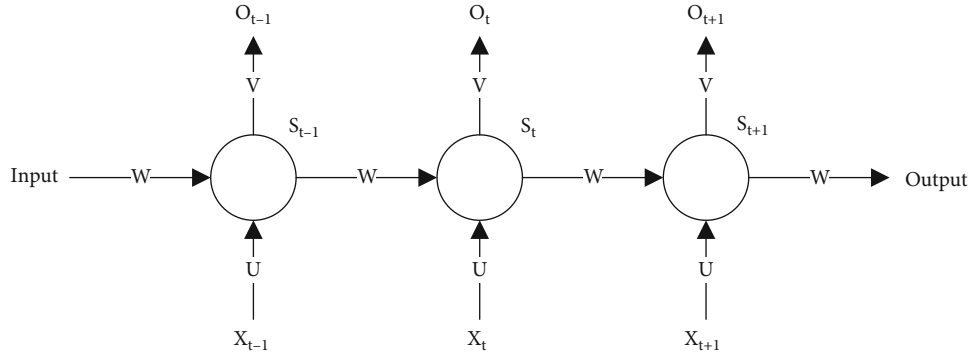


FIGURE 1: RNN pipeline.

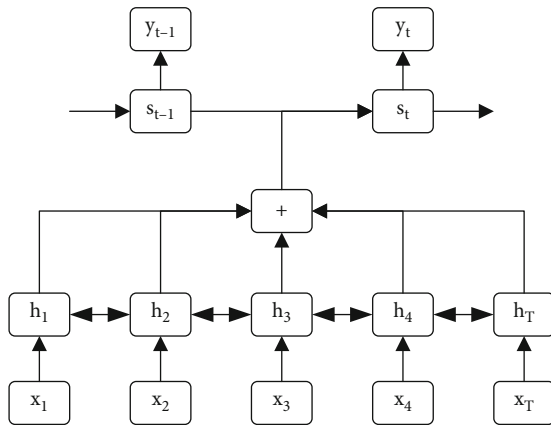


FIGURE 2: Attention in LSTM.

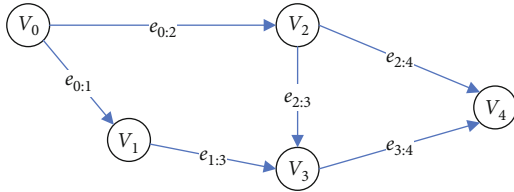


FIGURE 3: DAG structure.

represented by the current node to the next node, the two ends connected by the edge BB are used as the position representation.

3.3. ELMo-Based Dynamic Word Vector Embedding. The text characterization methods used by representative neural machine translation systems are all static word vectors, where each word corresponds to a fixed distributed representation. This fixed distributed representation cannot effectively represent the ambiguity of words. Whether for language, the phenomenon of polysemy or that a word expresses different meanings in different contexts is very common. In order to solve the problem of polysemy, researchers proposed a large-scale pretraining model represented by ELMo. Since it is pretrained on a large-scale corpus, when in use, the model is able to generate dynamic word vectors according to different downstream tasks.

Large-scale pretraining models and dynamic word vectors have been proposed for nearly two years. However, dynamic word vectors are rarely used as word embeddings in neural machine translation systems. This chapter proposes a multigranularity feature fusion method, which contains more information than the previous single-granularity input in the form of character granularity and multiple word granularity fusion. Therefore, this paper introduces the ELMo model as an additional encoding network to improve the encoding ability of the encoder and at the same time verifies the impact of dynamic word vectors on the performance of neural machine translation.

The ELMo model can perceive the context and generate dynamic word vectors. For the complex semantic and grammatical knowledge contained in words, the deep model structure of ELMo can effectively model. The process of generating dynamic word vectors by ELMo is divided into two stages. The first stage pretrains a deep Bi-LSTM language model, and the second stage extracts each state inside the hidden layer of the corresponding word in the pretraining model, and then generates the word vector through the transformation function. Figure 4 shows the structure of the ELMo model. The input of the ELMo model is a source language sentence that combines character granularity and multiple word granularities. Each character or word will obtain a dynamic word vector through the ELMo model.

3.4. Multigranularity Relative Position Encoding. In this chapter, a multigranularity position encoding approach based on relative position representation is proposed using the current relative position representation and multigranularity feature input. Absolute positioning is encoded using only one sequence in the original Transformer model. The principle that one input sequence corresponds to one position representation sequence is likewise applied to the relative position representation. However, there are two sequences of position representation in the multigranularity feature input that cannot be used directly with the existing relative representation. Using the head and tail node positions, the relative positions of words may be calculated from four different angles. After nonlinear transformation processing, the four generated relative distance matrices are spliced together to provide a multigranularity position encoding vector.

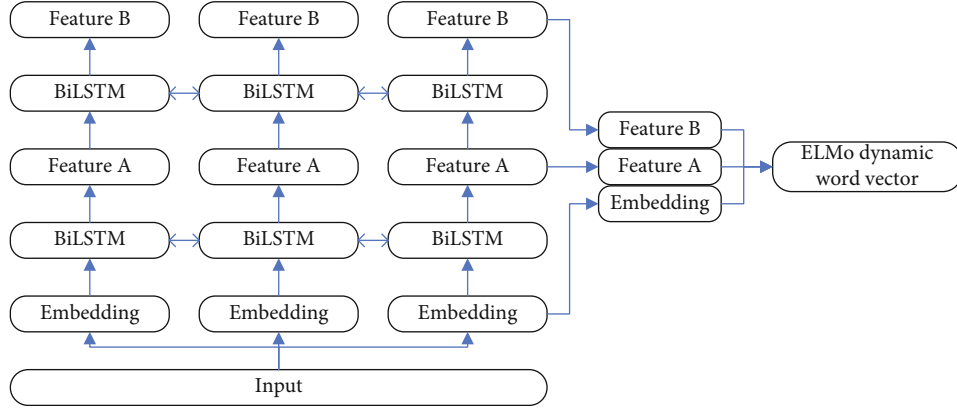


FIGURE 4: ELMo module.

For any word, there are head node position representation and tail node position representation. It is assumed that the relative position encoding vector between two words is to be computed under multigranularity features. First, calculate the distances between them from the four angles and get four relative distance matrices.

$$d_{ij}^{(hh)} = \text{head}(i) - \text{head}(j), \quad (6)$$

$$d_{ij}^{(ht)} = \text{head}(i) - \text{tail}(j), \quad (7)$$

$$d_{ij}^{(th)} = \text{tail}(i) - \text{head}(j), \quad (8)$$

$$d_{ij}^{(tt)} = \text{tail}(i) - \text{tail}(j). \quad (9)$$

First, the four relative distance matrices are spliced together, and then nonlinearly transformed, and finally the position encoding vector between them is obtained.

$$R_{ij} = \text{ReLU}\left(W_r \left(d_{ij}^{(hh)} \oplus d_{ij}^{(ht)} \oplus d_{ij}^{(th)} \oplus d_{ij}^{(tt)}\right)\right). \quad (10)$$

3.5. Multigranularity Self-Attention. This work adds the multigranularity relative position encoding to the self-attention mechanism, thereby modifying the original self-attention mechanism into a multigranularity self-attention mechanism. The self-attention mechanism of relative position encoding is

$$a_{ij} = \text{softmax}\left(x_i W_Q \left(x_j W_K + R_{ij}^K\right)^T\right), \quad (11)$$

$$o_i = \sum_j a_{ij} \left(x_j W_V + R_{ij}^V\right), \quad (12)$$

where x is feature and W is weight.

The relationship between relative position coding and relative position representation is

$$R_{ij}^K = W_r^K \text{PE}[\text{clip}(i - j, p_{\min}, p_{\max})], \quad (13)$$

$$R_{ij}^V = W_r^V \text{PE}[\text{clip}(i - j, p_{\min}, p_{\max})], \quad (14)$$

where W is weight.

Finally, the multigranularity self-attention mechanism proposed in this chapter draws on Equations (11) and (12) as the basis, and adopts a multigranularity form for the calculation of relative position encoding.

$$R_{ij}^K = \text{ReLU}\left(W_r^K \left(\text{PE}_{d_{ij}^{(hh)}} \oplus \text{PE}_{d_{ij}^{(ht)}} \oplus \text{PE}_{d_{ij}^{(th)}} \oplus \text{PE}_{d_{ij}^{(tt)}}\right)\right), \quad (15)$$

$$R_{ij}^V = \text{ReLU}\left(W_r^V \left(\text{PE}_{d_{ij}^{(hh)}} \oplus \text{PE}_{d_{ij}^{(ht)}} \oplus \text{PE}_{d_{ij}^{(th)}} \oplus \text{PE}_{d_{ij}^{(tt)}}\right)\right), \quad (16)$$

where ReLU is activation function.

3.6. Multigranularity Feature Combined with Dynamic Word Vector Model. The neural machine translation model of multigranularity features combined with dynamic word vectors (MGF-DWV) proposed in this chapter is shown in Figure 5. The model consists of four parts, which are multigranularity feature fusion part (MGFF), multigranularity relative position encoding part (MGRPE), dynamic word vector generation part (DWVG), and encoder-decoder part.

The encoder-decoder part uses the Transformer-based model as the backbone network, including 6-layer encoder, 6-layer decoder, and an output layer. The encoder consists of two sublayers, a multigranularity self-attention sublayer and a position feed-forward sublayer, each of which also has residual connections and layer normalization calculations. The decoder is the same as the original Transformer decoder, consisting of a multihead attention sublayer, an encoder-decoder attention sublayer, and a position feedforward sublayer. Each sublayer also has residual connections and layer normalization calculations, and the output layer is composed of linear transformation and Softmax layer.

The workflow of the model is as follows. In the first step, the text input sequence is divided into multiple granularities. Multigranularity includes character granularity and three-word segmentation granularities based on MSR, PKU, and CTB and is modeled by DAG fusion. In the second step, a directed acyclic graph is used to convert the multigranularity

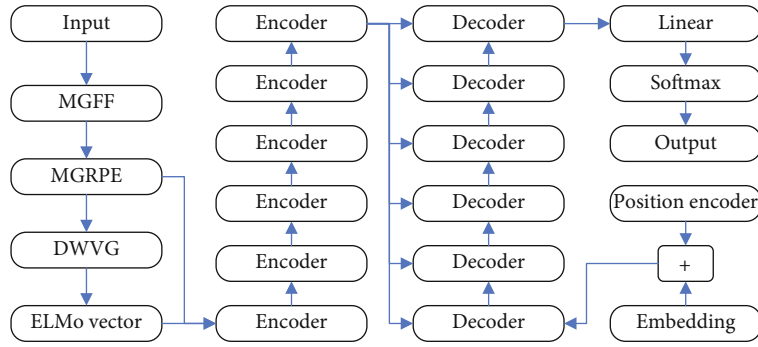


FIGURE 5: MGF-DWV pipeline.

representation of the input sequence into a sequence of granularity and corresponding position representations. The third step is to input the granularity sequence into the ELMo pretraining model to obtain dynamic text feature vectors. The fourth step is to convert the sequence of position representations into multigranularity relative position representations, which are further converted into relative position codes through trigonometric function position coding. In the fifth step, the dynamic text feature vector is loaded into the model encoder. After the multigranularity self-attention layer, the relative position encoding obtained in the previous step is integrated into the multigranularity self-attention layer. After obtaining the new text feature vector, the input position feed-forward layer sublayer is then passed to the subsequent encoder layer from bottom to top. In the sixth step, the output of the last encoder layer is passed as input to the encoder-decoder attention layer in the corresponding decoder. The translated result is used as input after word embedding and position encoding, and then passes through the self-attention layer and position feed-forward layer, bottom-up, and passed through the decoder. In the seventh step, the text feature vector finally calculated by the decoder is input to the output layer composed of the linear transformation layer and the Softmax layer, and the corresponding output result is obtained.

3.7. Online Translation System Based on Wireless Network. The goal of this work is to build an online translation platform that integrates machine learning and network, and this system adopts B/S structure. The basic functions should include task sending and receiving, machine translation, translation memory translation, manual proofreading, and corpus management. In addition, a task management system is required to record tasks and handle data transmission between subsystems. The main business process of the system is as follows: the user submits the text and document files that need to be translated to the system based on the wireless network through page input, file upload, etc. Documents and texts in various formats are uniformly converted into TXT format documents and submitted to an automatic translation system. The MGF-DWV automatic translation system first uses translation memory and translation engine to generate preliminary results for the original text submitted. The machine translation result is returned to the user through the wireless network, or the final result is sent to

the user after manual proofreading, and the manual modification result will be saved in the corpus for processing.

The system provides online translation in the form of web pages based on wireless network, and users can submit text, text documents, and URL addresses. For the window submission method, the user will get the result after filling in the translation content and submitting it. For the file upload method, the user downloads the translated file after submitting the file. For the web method, the user submits a URL to get the translated page. The task receiving module receives the original text in different forms through different interfaces and converts the original text of the task by calling the text format conversion module into the original text in TXT format and transmits it to the task management system. The task submission is a process of human-computer interaction, and the task submitter needs to be able to configure the translation requirements appropriately. Therefore, the focus of the design is to provide users with a convenient interface for attribute selection and task submission.

An automatic translation system consists of a translation memory and a machine translation engine. The task management system saves the translation content of the task receiving module as a TXT file and submits it to the translation model through the interface and sets the waiting threshold. If the system times out, the automatic translation process will be closed and an error will be reported. The automatic translation system outputs the result with the mark format, and the result includes the original text, mark, and translation. The identification is mainly to distinguish the sentences of the translation, the sentences in the memory, and the sentences in automatic translation. The output of the automatic translation is also in the form of TXT files. The automatic translation system provides translators with automatic translation results divided into sentences, indicating whether it is a memory matching result or a machine translation result.

The translation results obtained by the manual proofreading system from the automatic translation system are displayed on the manual proofreading page. The display method is the contextual comparison between the original text and the translated text. The original text cannot be modified, but only the translated text can be modified. The manual proofreading part is dominated by translators, and the computer is only used as a recording and auxiliary tool. The design of this part focuses on how to provide translators

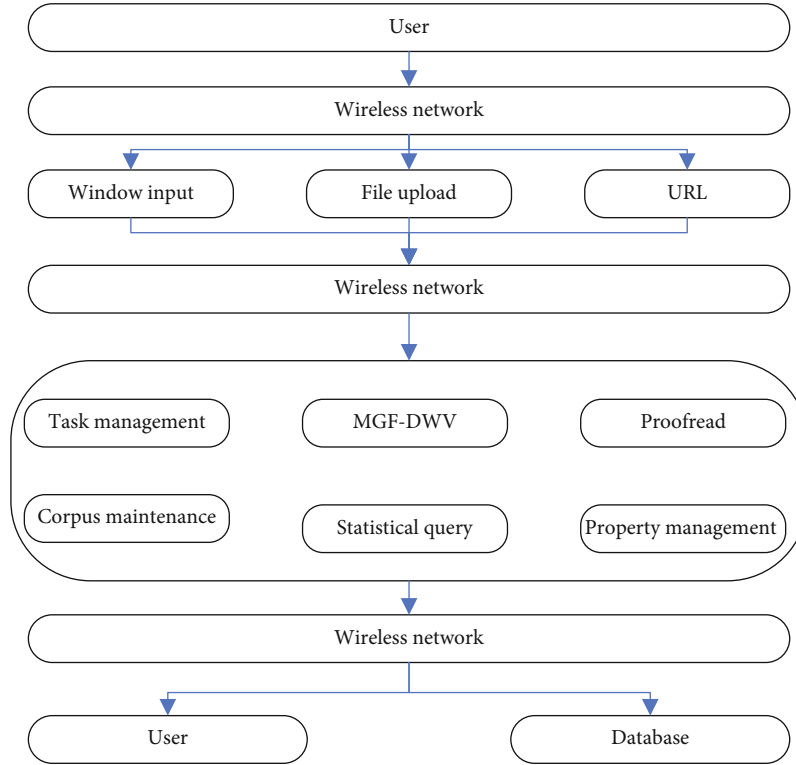


FIGURE 6: Online translation system pipeline.

TABLE 1: The dataset information.

Dataset	Training set	Test set	Total
TDA	50281	15017	65298
TDB	71935	30125	102060

with a friendly human-machine interface, convenient translation aids, and finally record and save the translators' work.

The task management system runs through the whole world and starts task recording whenever a new task is initiated until the task ends. First, record the original text submitted by the user, and then record the automatic translation result. If the automatic translation times out, it will be transferred to the manual proofreading system or the translation failure information will be returned to the user. If manual intervention is required, the results will be submitted to manual proofreading, and the manual proofreading results will be recorded. Record the success or failure of the task, save the entire task record into the database for query, and finally record the end time of the task. The task management system controls the operation of the entire system.

The corpus and translation memory are independent, and the corpus is regularly reviewed by the maintainers. Periodically updated results are imported from the corpus into the main memory by the system maintainer, and the corresponding content in the corpus is deleted at the same time.

The statistical query includes daily statistical query and cumulative statistical query, wherein the daily statistical query is to record the usage status of the system. It needs

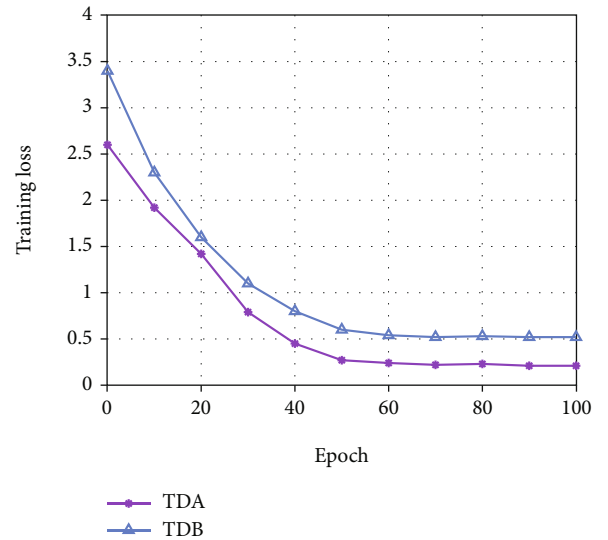


FIGURE 7: The training loss of MCF-DWV on TDA and TDB.

TABLE 2: Comparison with other machine learning method.

Method	TDA	TDB
LSTM	23.8	26.9
Bi-LSTM	26.1	30.3
Transformer	28.5	33.9
MCF-DWV	31.6	35.7

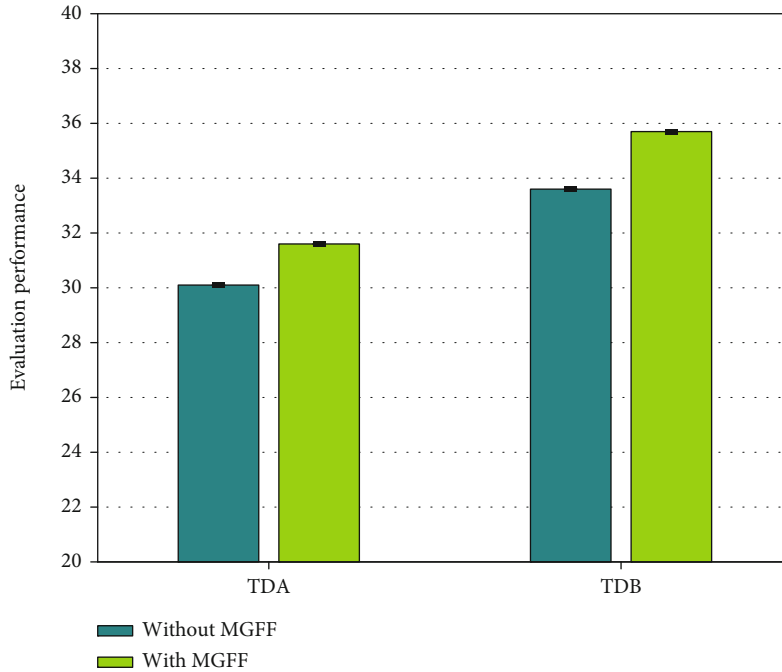


FIGURE 8: Evaluation of multigranularity feature fusion.

to count the number of translated bytes, the way the translation request occurs, the time it takes to complete the task, the data flow and task volume in each period, and the success rate of memory matching.

To facilitate the recording of tasks, the attributes of tasks need to be managed. Relevant attributes include submission method, submission time, translation direction, return method, return time, whether manual intervention is required, and professional field division. In addition, in order to personalize the configuration of the system, the attributes of the automatic translation system's memory matching degree, automatic translation limited time, and task timeout information can be set through the page.

The online translation system that integrates wireless network and machine learning proposed in this work is demonstrated in Figure 6.

4. Experiment

4.1. Evaluation on MCF-DWV. This work is based on the crawler technology to collect the corresponding data from the network to train and test the network. The data used come from two datasets, and the specific information is demonstrated in Table 1. The evaluation metric utilized in this work is BLEU.

First, this work analyzes the training loss of MCF-DWV on TDA and TDB dataset, because network training is the basis for subsequent testing, as demonstrated in Figure 7.

With the deepening of training, the loss on the two datasets gradually decreases, and finally convergence is achieved.

Then, the proposed MCF-DWV is compared with other machine learning translation methods, including LSTM, Bi-LSTM, and Transformer, and the comparison results are demonstrated in Table 2.

As shown in the data in the table, the translation performance obtained by the LSTM network is the lowest, and the use of Bi-LSTM and Transformer strategies brings a certain degree of improvement. However, when using the MCF-DWV proposed in this work, the highest BLEU can be obtained, which significantly outperforms other methods.

MCF-DWV utilizes multigranularity feature fusion (MGFF). To verify the superiority of this feature fusion strategy, the translation performance with MGFF and without MGFF is compared, and the comparison data is demonstrated in Figure 8.

After using the MGFF strategy, the BLEU on the two datasets is improved by 1.5% and 2.1%, respectively, which confirms the superiority of the MGFF strategy designed in this work.

MCF-DWV utilizes dynamic word vector generation (DWVG). To verify the superiority of this dynamic word vector generation mechanism, the translation performance with DWVG and without DWVG is compared, and the comparison data is demonstrated in Figure 9.

After using the DWVG strategy, the BLEU on the two datasets is improved by 1.7% and 1.6%, respectively, which confirms the superiority of the DWVG strategy designed in this work.

MCF-DWV utilizes multigranularity relative position encoding (MGRPE). To verify the superiority of this position encoding strategy, the translation performance with different position encoding strategy is compared, and the comparison data is demonstrated in Figure 10.

As demonstrated in the data comparison in the figure, the translation performance corresponding to the multigranularity relative position encoding method is the highest. Compared with absolute position encoding and relative position encoding, BLEU is improved to different degrees.

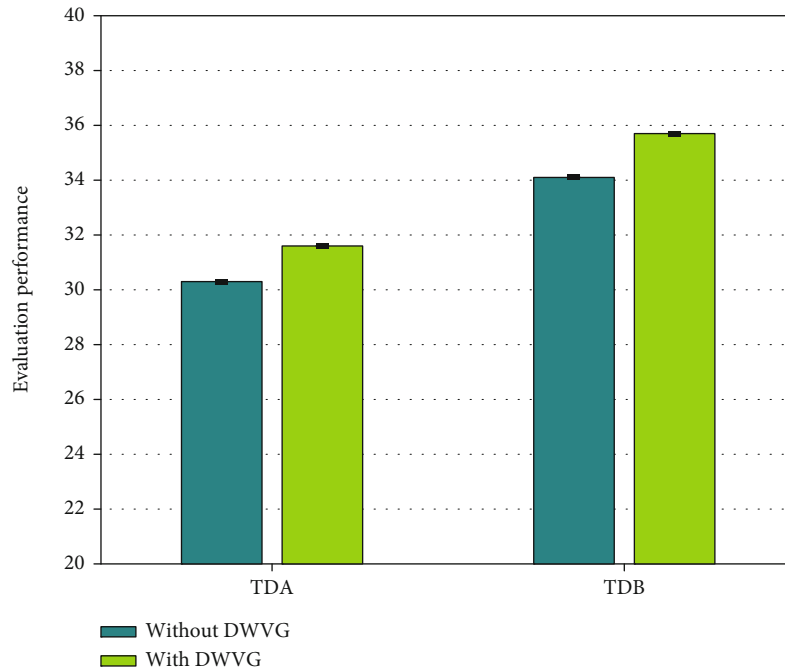


FIGURE 9: Evaluation of dynamic word vector generation.

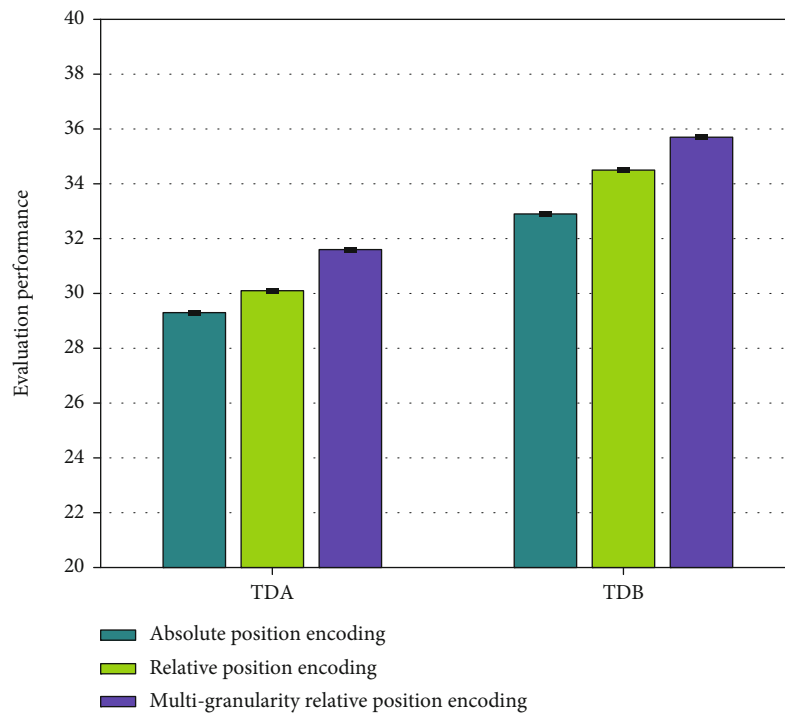


FIGURE 10: Evaluation of multigranularity relative position encoding.

MCF-DWV utilizes multigranularity self-attention (MGSA). To verify the superiority of this attention strategy, the translation performance with different self-attention strategy is compared, and the comparison data is demonstrated in Figure 11.

As demonstrated in the data comparison in the figure, compared with the traditional self-attention mechanism, after using the multigranularity self-attention strategy, the BLEU on the two datasets is improved by 1.4% and 1.1%, respectively.

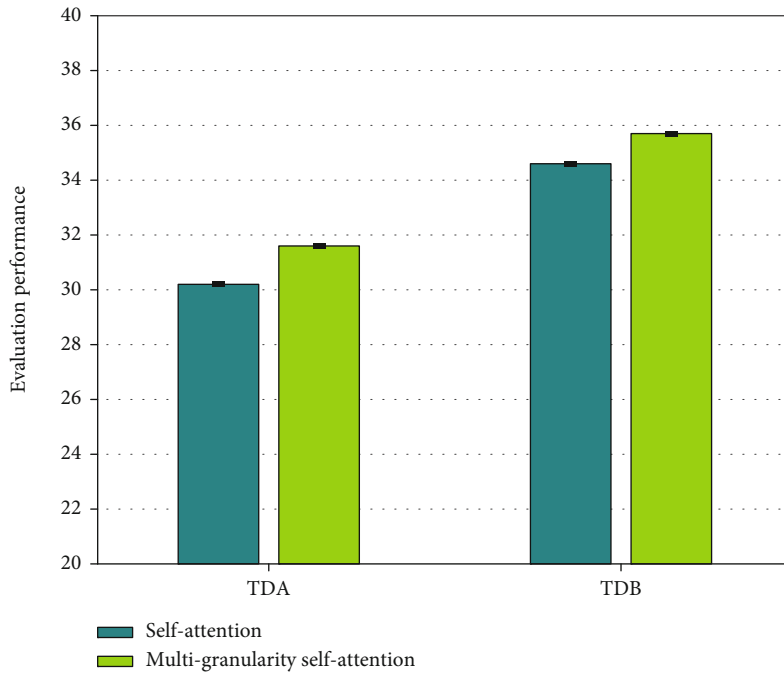


FIGURE 11: Evaluation of multigranularity self-attention.

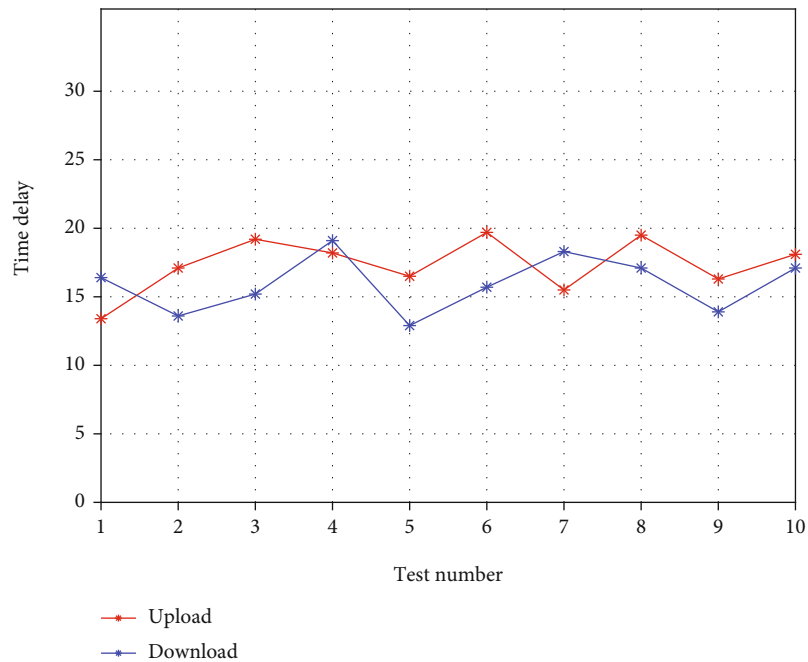


FIGURE 12: Evaluation of time delay.

4.2. *Evaluation on Online Translation System.* The online translation system is designed based on machine learning and wireless network and is used to upload the content to be translated, and then download the translated content. Therefore, the requirements for the system delay of uploading and downloading are very important. This work compares the delays of these two stages, respectively, and the results are demonstrated in Figure 12.

As illustrated in the delay data in the table, in the process of uploading and downloading, the delay of the system is relatively low, which is controlled within 20 ms.

In addition to the delay, the success rate of content upload and download is also a very important indicator. This work also conducted 10 tests to analyze the upload and download success rates. The results are demonstrated in Table 3.

TABLE 3: Evaluation of system success rate.

Test	1	2	3	4	5	6	7	8	9	10
Upload	99.6	100	100	99.8	99.9	100	100	100	99.7	100
Download	100	100	99.8	100	99.9	100	99.6	100	100	99.9

As demonstrated in the table, the online translation system proposed in this work has a high success rate in the uploading and downloading process, both above 99.5%, which verifies the reliability of the system.

5. Conclusion

With the rapid development of artificial intelligence technology, machine translation technology has also received extensive attention from researchers. Neural machine translation technology has a simple translation model, strong operability, and does not require a lot of expert knowledge and has become the mainstream translation model. However, it still faces many problems. On the one hand, because it relies on neural networks to be trained on large-scale corpora, the model training period is very long. On the other hand, the translation effect of the model is still unsatisfactory, and there will be omissions and mistranslations. Despite the fact that natural language processing is often referred to as AI's crown jewel, understanding and processing natural language remains a challenge. Machine translation's representative application is still lacking in quality. Machine translation has unquestionably made significant strides forward thanks to developments in deep learning and machine learning. This work applies machine learning and wireless network technology to build an online translation system for real-time translation. First, this work proposes a multi-granularity feature fusion method based on a directed acyclic graph, which uses a directed acyclic graph to fuse different granularities as input and obtain a position representation. Secondly, this paper improves the Transformer model and proposes multigranular position encoding and multigranular self-attention. Then, on the basis of multigranularity features as input, this work introduces dynamic word vectors to improve the word embedding module and uses the ELMO model to obtain dynamic word vector embeddings. Finally, this work builds a multigranularity feature-dynamic word vector machine translation model with above strategy, deploys it on the server. Users can upload the content to be translated and download the translated content through the wireless network and realize an online translation system based on machine learning and wireless network. This work conducts systematic experiments on the proposed translation model and translation system. The experimental results verify that the accuracy of the translation model is high enough, and the designed translation system also has low latency and stability.

Data Availability

The datasets used during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest

The author declares that he has no conflict of interest.

References

- [1] F. Stahlberg, "Neural machine translation: a review," *Journal of Artificial Intelligence Research*, vol. 69, pp. 343–418, 2020.
- [2] R. Dabre, C. Chu, and A. Kunchukuttan, "A survey of multilingual neural machine translation," *ACM Computing Surveys (CSUR)*, vol. 53, no. 5, pp. 1–38, 2020.
- [3] Y. Liu, J. Gu, N. Goyal et al., "Multilingual denoising pre-training for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.
- [4] J. Du, C. Jiang, Z. Han, H. Zhang, S. Mumtaz, and Y. Ren, "Contract mechanism and performance analysis for data transaction in mobile social networks," *IEEE Transactions on Network Science and Engineering*, vol. 6, no. 2, pp. 103–115, 2019.
- [5] X. Pan, M. Wang, L. Wu, and L. Li, "Contrastive learning for many-to-many multilingual neural machine translation," <https://arxiv.org/abs/2105.09501>.
- [6] L. Song, D. Gildea, Y. Zhang, Z. Wang, and J. Su, "Semantic neural machine translation using AMR," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 19–31, 2019.
- [7] S. A. Busari, K. M. S. Huq, S. Mumtaz et al., "Generalized hybrid beamforming for vehicular connectivity using THz massive MIMO," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 9, pp. 8372–8383, 2019.
- [8] M. Fomicheva, S. Sun, L. Yankovskaya et al., "Unsupervised quality estimation for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 539–555, 2020.
- [9] L. Zhou, J. Zhang, and C. Zong, "Synchronous bidirectional neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 91–105, 2019.
- [10] U. V. Ucak, I. Ashyrmamatov, J. Ko, and J. Lee, "Retrosynthetic reaction pathway prediction through neural machine translation of atomic environments," *Nature Communications*, vol. 13, no. 1, pp. 1–10, 2022.
- [11] M. Tufano, C. Watson, G. Bavota, M. D. Penta, M. White, and D. Poshyvanyk, "An empirical study on learning bug-fixing patches in the wild via neural machine translation," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, vol. 28, no. 4, pp. 1–29, 2019.
- [12] J. E. Ortega, R. Castro Mamani, and K. Cho, "Neural machine translation with a polysynthetic low resource language," *Machine Translation*, vol. 34, no. 4, pp. 325–346, 2020.
- [13] S. Lin, B. Zheng, G. C. Alexandropoulos, M. Wen, F. Chen, and S. Mumtaz, "Adaptive transmission for reconfigurable intelligent surface-assisted OFDM wireless communications," *IEEE*

- Journal on Selected Areas in Communications*, vol. 38, no. 11, pp. 2653–2665, 2020.
- [14] Y. Belinkov, N. Durrani, F. Dalvi, H. Sajjad, and J. Glass, “On the linguistic representational power of neural machine translation models,” *Computational Linguistics*, vol. 46, no. 1, pp. 1–52, 2020.
- [15] S. Stoll, N. C. Camgoz, S. Hadfield, and R. Bowden, “Text2-Sign: towards sign language production using neural machine translation and generative adversarial networks,” *International Journal of Computer Vision*, vol. 128, no. 4, pp. 891–908, 2020.
- [16] J. Zhang and C. Zong, “Deep neural networks in machine translation: an overview,” *IEEE Intelligent Systems*, vol. 30, no. 5, pp. 16–25, 2015.
- [17] R. Cao, C. Freitas, L. Chan, M. Sun, H. Jiang, and Z. Chen, “ProLanGO: protein function prediction using neural machine translation based on a recurrent neural network,” *Molecules*, vol. 22, no. 10, p. 1732, 2017.
- [18] H. Mi, B. Sankaran, Z. Wang, and A. Ittycheriah, “Coverage embedding models for neural machine translation,” <https://arxiv.org/abs/1605.03148>.
- [19] D. He, Y. Xia, T. Qin et al., “Dual learning for machine translation,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.
- [20] M. L. Forcada, “Making sense of neural machine translation,” *Translation Spaces*, vol. 6, no. 2, pp. 291–309, 2017.
- [21] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, “Deep recurrent models with fast-forward connections for neural machine translation,” *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 371–383, 2016.
- [22] P. Shaw, J. Uszkoreit, and A. Vaswani, “Self-attention with relative position representations,” <https://arxiv.org/abs/1803.02155>.
- [23] B. Zhang, D. Xiong, and J. Su, “Accelerating neural transformer via an average attention network,” <https://arxiv.org/abs/1805.00631>.
- [24] K. Chen, R. Wang, M. Utiyama, E. Sumita, and T. Zhao, “Neural machine translation with sentence-level topic context,” *IEEE/ACM Transactions on Audio, Speech, Language Processing*, vol. 27, no. 12, pp. 1970–1984, 2019.
- [25] J. Guo, X. Tan, D. He, T. Qin, L. Xu, and T.-Y. Liu, “Non-autoregressive neural machine translation with enhanced decoder input,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3723–3730, 2019.
- [26] X. X. Wang, C. H. Zhu, S. Li, T. J. Zhao, and D. Q. Zheng, “Neural machine translation research based on the semantic vector of the tri-lingual parallel corpus,” in *2016 International Conference on Machine Learning and Cybernetics (ICMLC)*, pp. 69–74, Jeju, Korea (South), 2016.
- [27] N. Akoury, K. Krishna, and M. Iyyer, “Syntactically supervised transformers for faster neural machinetranslation,” <https://arxiv.org/abs/1906.02780>.
- [28] A. Vaswani, N. Shazeer, N. Parmar et al., “Attention is all you need,” *Advances in Neural Information Processing Systems*, vol. 30, 2017.
- [29] L. M. Werlen, N. Pappas, D. Ram, and A. Popescu-Belis, “Self-attentive residual decoder for neural machine translation,” <https://arxiv.org/abs/1709.04849>.
- [30] H. Li, *Deep Learning for Natural Language Processing: Advantages and Challenges*, National Science Review, 2017.
- [31] C. Shao, Y. Feng, J. Zhang, F. Meng, X. Chen, and J. Zhou, “Retrieving sequential information for non-autoregressive neural machine translation,” <https://arxiv.org/abs/1906.09444>.
- [32] C. Shao, J. Zhang, Y. Feng, F. Meng, and J. Zhou, “Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 198–205, 2020.
- [33] J. Zhang, H. Luan, M. Sun et al., “Improving the transformer translation model with document-level context,” <https://arxiv.org/abs/1810.03581>.
- [34] S. Ma, D. Zhang, and M. Zhou, “A simple and effective unified encoder for document-level machine translation,” in *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 3505–3511, 2020.
- [35] V. Macé and C. Servan, “Using whole document context in neural machine translation,” <https://arxiv.org/abs/1910.07481>.