Hindawi
Mathematical Problems in Engineering
Volume 2023, Article ID 9782682, 1 page
https://doi.org/10.1155/2023/9782682

Retraction

Retracted: Research on the Construction of a Bidirectional Neural Network Machine Translation Model Fused with Attention Mechanism

Mathematical Problems in Engineering

Received 1 August 2023; Accepted 1 August 2023; Published 2 August 2023

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope
(2) Discrepancies in the description of the research reported
(3) Discrepancies between the availability of data and the research described
(4) Inappropriate citations
(5) Incoherent, meaningless and/or irrelevant content included in the article
(6) Peer-review manipulation

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] G. Zuo, "Research on the Construction of a Bidirectional Neural Network Machine Translation Model Fused with Attention Mechanism," *Mathematical Problems in Engineering*, vol. 2022, Article ID 2971876, 11 pages, 2022.

Hindawi
Mathematical Problems in Engineering
Volume 2022, Article ID 2971876, 11 pages
https://doi.org/10.1155/2022/2971876

Research Article

Research on the Construction of a Bidirectional Neural Network Machine Translation Model Fused with Attention Mechanism

Guangming Zuo

Huaiying Institute of Technology, Huaian, Jiangsu 223000, China

Correspondence should be addressed to Guangming Zuo; zuoguangming@hyit.edu.cn

Received 23 June 2022; Revised 6 July 2022; Accepted 25 July 2022; Published 19 August 2022

Academic Editor: Zaoli Yang

With the development of deep learning, neural machine translation has also been paid attention and developed by researchers. Especially in the application of encoder-decoder in natural language processing, the translation performance has been significantly improved. In 2014, the attention mechanism was used in neural machine translation, the performance of translation was greatly improved, and the interpretability of the model was increased. This research proposes a research idea of sparsemax combined with AAN machine translation model and conducts multiple ablation experiments for experimental verification. This chapter first studies the problem of insufficient sparse normalization when generating target words in the attention mechanism and studies the neural machine translation model incorporating the sparse normalization calculation method. It solves the problem of inductive bias in the data transfer process of related sub-layers in the model. By combining the strategy of sparse normalization, the similarity value of related word vectors can be obtained more accurately when aligning words, which is more convenient for this chapter. Calculate and analyze the specific principles of the model. In addition, when the model faces a large vocabulary in the decoding stage, too many weights of scattered vocabulary vectors are not conducive to the generation of correct target values. After using the sparse normalization strategy, it can reduce the number of inconveniences. The calculation between related words optimizes the classification accuracy of the target vocabulary. In this chapter, aiming at the waste of the transformer's decoder calculation in the inference stage, the average attention structure is used to replace the attention calculation layer of the first layer of the decoder part of the original model. Each moment is only related to the previous moment, which alleviates the waste of computing resources.

1. Introduction

Translation is the first step in all language and text communication, the most basic and the most important step. Translation affects not only the efficiency of communication, but also the possibility of its breadth and depth. However, translation is also a time-consuming and expensive activity, in both time and money. From a personal point of view, it is very difficult to master multiple languages in a short lifetime. From the national and collective point of view, professional translators are in short supply and limited. Therefore, translation with high quality and quantity directly affects the dissemination of effective information and the current process of globalization. With the introduction of information theory and the continuous development of computer technology, concepts such as artificial intelligence, natural language processing, and neural networks were put forward in the 1950s. People increasingly hope to use advanced computer technology to solve the problem of language. Under this wave, in a letter written by Warren Weaver in March 1947, he proposed the idea of using electronic computers to translate human natural language documents. Subsequently, countless attempts and research have been put into this field. For decades, machine translation has grown rapidly and has become an integral part of the translation industry. So far, machine translation is still an important research topic in universities and laboratories around the world, and machine translation has begun to be applied in industry. Machine translation technology has experienced several stages, such as the initial rule-based machine translation, the development of statistics-based machine translation, and the current neural network-based machine translation. Despite the considerable development, there is still a big gap between the effect achieved by machine translation and the translation effect in the ideal state of human beings. This field still needs continuous attention and research. It is believed that with the continuous development of technology, machine translation will play an increasingly important role in the translation industry and even other natural language processing fields.

Traditional neural machine translation is based on the Seq2Seq framework, utilizing two recurrent neural network models (recurrent neural network, RNN), one of the two RNNs is used to solve the source language used to encode to generate a constant-size intermediate vector, and then another RNN is used to decode the intermediate vector to generate the target sequence. After improvement, Bahdanau et al. added the attention mechanism to neural machine translation. This also lays a valuable cornerstone for the widespread application of attention mechanisms in natural language processing [1–8]. This paper mainly conducts research on neural machine translation-related content for the transformer model, which is now brilliant in various fields involved in deep learning. Research on neural network translation model based on transformer-based sparse normalization operation aims at the maximum value of sparse classification vector in the multi-classification problem of softmax in existing machine translation to optimize the experimental effect. Aiming at the role of the overall structure of the model in translation, combined with the idea of deliberating neural network, the translation model of transformer is studied.

2. Related Work

Neural Machine Translation is also known as "Neural Machine Translation“ (NMT). Its basic idea is to map the source language sentence sequence to the target language sentence sequence through the neural network structure. In the neural network structure, the continuous vector representation is used to model the translation process, so it has better generalization performance and at the same time avoids the problem of too strong independence assumption in traditional machine translation. Statistical machine translation uses discrete symbols to convert discrete symbols, while neural network machine translation first converts discrete symbols into continuous vectors and then converts them into discrete symbols. This is the essential difference between statistical machine translation and neural machine translation. In the initial machine translation based on the Seq2Seq framework, the encoder generates a context vector C of invariant size, and the decoder uses the fixed-length vector as the decoder input and outputs the final answer sequence. Usually, both encoders and decoders are trained by maximizing the maximum likelihood estimate for each pair of input spatial sequence and output spatial sequence. There is an obvious defect. When the dimension of the output fixed-length context vector of the encoder is too low, and the translation input sentence is too long, it is difficult to express all the input information. Therefore, Bahdanau et al. invented a variable context vector calculation method, and they brought the attention mechanism into the model, so that the model can select more relevant inputs, which improves the model's ability to distinguish and translate overall performance. On the basis of Bahdanua et al., Luong et al. of Stanford University proposed various variants of the attention mechanism. For example, only the vector of the hidden layer is used in the calculation, and an instant positioning strategy is introduced, so that the model has a local ability to pay attention. Gehring et al. of Facebook introduced a simple encoder framework for neural translation based on convolutional networks. This approach is more parallelizable than recurrent networks and provides a shorter path to capture long-term dependencies in the source. They used source location embeddings and CNNs with kernels of different sizes for attention score computation and input aggregation. Their experiments show that convolutional encoders perform on par or better than the baseline relative to bidirectional LSTM encoders. Then, Gehring et al. combined the gated convolutional network and multi-step attention mechanism proposed by Dauphin et al., and invented a neural translation based on convolutional neural network for both encoder and decoder models, which performs better on large benchmark datasets and is an order of magnitude faster. Compared to recurrent networks, their convolutional strategy is able to find constituent structures in sequences more easily because the representation is structured hierarchically. In 2017, Google's Gehring et al. [9] invented a new and simple network framework. The model is only based on attention mechanism and feedforward network, and does not use a little RNN and CNN structure. The experimental results show that transformer's translation effect is more competitive and has higher parallelism; in addition, it significantly reduces the time required for model training. Transformer uses a combination of multi-head self-attention structure and feedforward network to construct a neural machine translation model for encoding and decoding, which has caused great interest in the field of machine translation and even deep learning. However, the transformer model has some imperfections in its own framework. One is that the self-attention mechanism needs to calculate the mutual attention score between all sequences for the input sequence in each round, and its computational time complexity is the square of the input sequence. The input context of the model is a previously preset fixed length. In the fine-tuning stage, the model cannot obtain context dependencies that exceed the maximum length. Due to the truncation of long text and the fragment encoding of the split by the transformer, there is context fragmentation). In 2020, Kitaev et al. proposed the reformer model to improve large-scale transformers and mainly proposed two improvement strategies. First, they combined the locality-sensitive hashing approach to compute attention, reducing its computational complexity from 2O(L) to O(Llog L) (where L is the length of the input sentence); in addition, they proposed reversible residual layers to replace previous residuals only that need to be activated once at any stage in the training process, rather than stacking layers several times. In 2019, Dai et al. invented the transformer-XL model, which can learn contextual dependencies beyond the maximum length without damaging the temporal effect of sequences. It consists of two parts: segment-level recurrence mechanism and a relative positional encoding scheme. The neural network structure can not only capture the long-term dependency, but also solve the difficulty of fragmentation in the context [9–15].

3. Transformer-Based Machine Translation Model Research

At present, neural machine translation technology is developing rapidly, but there are still many problems to be solved and methods can be optimized and improved in the development process. This chapter first analyzes some problems existing in the previous neural machine translation and demonstrates the reliability of the research strategy of this chapter through experiments. Aiming at some shortcomings of softmax, a sparse normalization strategy is adopted to alleviate the problem of inductive bias between model data. In addition, this chapter adopts the average attention mechanism to alleviate the low decoding efficiency of the transformer model in the inference stage.

3.1. Overall Structure of the Model. Main framework used in this study, as shown in Figure 1, is an improved neural machine translation model based on the overall structure of transformer. For the input word vector, this chapter uses the common subword model to characterize the word vector.

A) Position encoding of word vectors: Because the model in this chapter does not have recursive and convolutional structures, the learning of relative position information by the model needs to be marked by adding position information to the sequence. The position information includes the relative position or absolute position information of the relevant input. The positional encoding used in this chapter continues the relatively simple sine positional encoding. As shown in formula (1), the encoding vector of the corresponding position is generated by using the sine and cosine functions with inconsistent frequencies, and finally the input word vector adds directly [16].

$$\overline{p_t}^{(i)} = \begin{cases} \sin(w_k * t), & \text{if } i = 2k, \\ \cos(w_k * t), & \text{if } i = 2k + 1, \end{cases} \tag{1}$$

where t is the position of the input and the frequency w_k is defined as follows:

$$w_k = \frac{1}{10000^{2k/d}}. \tag{2}$$

From the function definition, it follows that the frequency decreases along the vector dimension. All elements in the position code match a sinusoid. Therefore, it appears in a geometric progression from 2π to $10000 * 2\pi$ in wavelength. It can usually be seen $\overline{p_t}^{(i)}$ that the position code represents a sine and cosine pair containing each frequency (d is divisible by 2, and the embedding dimension of the model is generally consistent). In addition to the simplicity and simplicity of the method, the sine and cosine approach is used because it can handle sentences that are longer than the input sequence in training.

B) Attention mechanism and feedforward neural network layer: This study adopts the multi-head attention mechanism of scaling dot product. This mechanism is introduced in detail in the attention mechanism section of Chapter 2. Multi-head attention can help the model capture more sufficient features or information, which will not be repeated here. In addition to the attention layer, the position-wise feedforward neural network in the model also plays a crucial role, as shown in Figure 2, which consists of two fully connected feedforward networks, and the two networks handle all positions separately. The entire FFN is mainly composed of two different linear transformations connected by the ReLU activation function, as shown in [17]

$$\mathrm{FFN}(x) = \max(0, xw_1 + b_1)w_2 + b_2. \tag{3}$$

Although linear transformations are used in different places, the parameters are different everywhere. Here, FFN can also be achieved by convolution operations with convolution kernel size 1. The specific input and output use $d_{\mathrm{model}} = 512$, and the intermediate layer uses d ff = 2048. It can be seen from the expression that the feedforward neural network layer is essentially a series of two linear changes, but the fully connected layer in the middle equally processes each position in the sentence, which enhances the expressive ability of the model.

C) Residual network and normalization layer: As the number of layers of the network increases, the training effect of the model on the training set will become worse, because as the number of layers of the network increases, training and optimization become more and more difficult, resulting in the degradation problem of deep learning. He Kaiming's team invented the famous residual learning structure to simplify the training of deeper networks. They cleverly constructed the model sub-layer into a residual function that can learn the input of the reference layer, as shown in (4) and Figure 3, without learning the function that does not appear. This model has been fully verified in the literature, which proves that the residual network optimization is more convenient, and the accuracy is positively correlated with the depth; that is, the deeper the depth, the higher the accuracy [18].

$$F(X) = H(X) + X. \tag{4}$$

Since the data are passed through the model, it will go through multiple rounds of calculations, resulting in the output data being in the saturation range of the activation function. As the number of iterations increases, the parameters between the layers in the network are also updated and changed. The following two situations are mainly caused: first, when there are slight changes in the weights in the lower layers of the network, because linear transformations and nonlinear activation mappings exist in all network layers, these slight changes will be caused by the increase in the number of network layers. Secondly, the input distribution of any layer will change as the

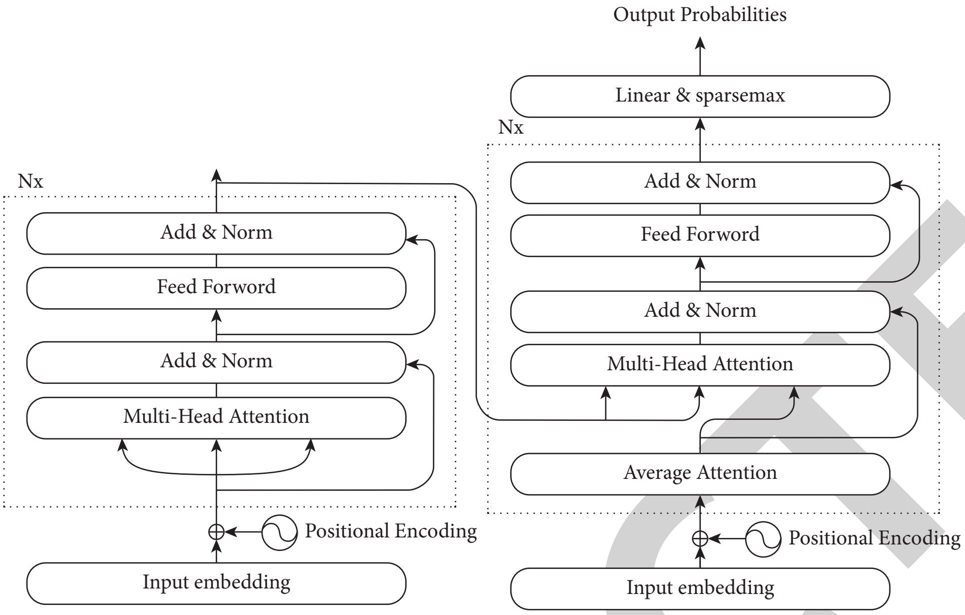


FIGURE 1: Schematic diagram of the overall framework of the model.

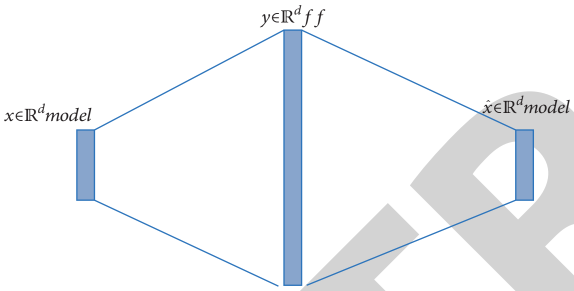


FIGURE 2: Schematic diagram of 2-position feedforward neural network.

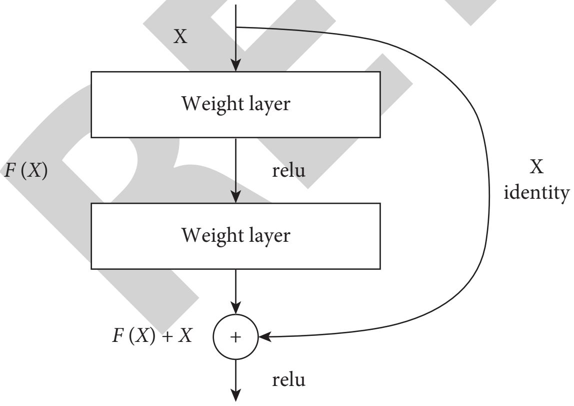


FIGURE 3: Schematic diagram of residual network.

model parameters change, so it is difficult for the deep network to match these corresponding distributions, which eventually makes it difficult for the model to converge quickly during training. Academics call the above phenomenon as internal covariate shift.

Therefore, it is necessary to calculate the normalization calculation by calculating the mean and variance for each sample of each sub-layer of the model, rather than calculating the mean and variance of all batch directions. The specific mathematical formula is shown in [19]

$$LN\left(x_i\right) = \alpha \times \frac{x_i - \mu}{\sqrt{\partial^2 + \varepsilon}}. \tag{5}$$

Among them, μ represents the mean and ∂ represents the variance. $\mu = (1/T)\sum_i x_i$. (remove the root sign). $\partial = \sqrt{1/T\sum_i \left(x_i - \mu\right)^2}$, formula (4).

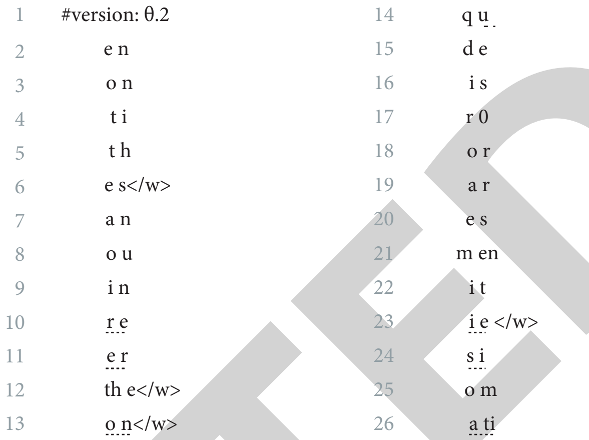
D) Linear connection and softmax layer: The output of the decoder part is a vector represented by floating point numbers, which is projected into a vector of the size of the target language vocabulary through a linear fully connected network layer. In this chapter, we use a subword encoding vocabulary of size 40,000 dimensions. Each element in the projected vector uniquely represents the score of a word. After the linear layer, the transformer uses the softmax layer, which converts all the scores into the numerical distribution of the vocabulary size through softmax, picks out the index number with the highest value, and then uses the index number to match the corresponding word in the vocabulary output. Since the softmax normalization calculation makes all probabilities greater than zero, such as −310, and we want to get a more sparse output, it is advantageous to have a larger probability [20–23].

3.2. Model Construction Based on Transformer. This study will mainly analyze the main framework of the model in this chapter, as well as the specific optimization strategy and the effect comparison of related ablation experiments. The main operation of the main body design and block optimization of the model is to improve the encoder-decoder model based on the attention mechanism. Finally, the reliability of the model is demonstrated by designing experiments. This research first introduces the word vector representation method of BPE and then introduces the average attention network and sparse normalization function in detail.

3.2.1. Word Vector Representation Based on BPE. Nowadays, the subword algorithm has gradually become an important strategy for improving the performance of various models in natural language processing. Since BERT was proposed in 2018, it has swept the major rankings of the natural language processing industry, various pretraining models have emerged, and the word algorithm has become the basic configuration of major algorithm models. Therefore, in the research of machine translation model, word algorithm is very important. The following section will focus on the relevant content. In terms of data compression, the simple method of byte pair (BPE) or binary encoding is usually used. In short, it is to replace the consecutive bytes with the highest probability of occurrence with other character representations that do not exist in the data. The original data are required later and generated by querying and replacing the vocabulary. OpenAI GPT-2 and Facebook RoBERTa are also vocabularies established through this type of strategy.

The main advantages of this method are as follows: the method is very effective in balancing the number of characters required in sentence encoding with the size of the vocabulary of the corpus. However, this strategy also has an unavoidable disadvantage; that is, it is impossible to obtain multi-shard output with probability. The main reason is that the core of this method is greedy algorithm and fixed sign exchange. The steps of the specific algorithm example are as follows: (1) first, we need to collect a large enough training corpus; (2) next, we need to determine the size of the required word list; (3) next, all the words are split into character-level sequences, and "$</w>$" is added to the end of each sequence to calculate the frequency of occurrence of all words. Words at this stage are character-based. For example, the word frequency of "word" is 10, so we can express it as "word $</w>$": 10; (4) calculate the number of occurrences of all consecutive characters, and combine the highest frequency characters into a new word character. Finally, the abovementioned step 2 is repeated until the fourth step is reached, until the word expression reaches the set value, or the number of occurrences of the byte pair with the most number of occurrences is 1. The main use of "$</w>$" is to indicate that the word symbol is the suffix of the whole word. For example, the word "es" can appear in other parts without adding "$</w>$," such as "es si e." The addition of "$</w>$" indicates that the character is at the end of the word, such as "stu di es$</w>$," and the two meanings are completely different. Part of the vocabulary in this chapter is shown in Figure 4.

After each vocabulary merging operation, the following three situations will occur: (1) the number of words increases, which means that after adding new words, the original two subwords will still exist (i.e., the two words do not appear consecutively). (2) The number of words remains the same, which means adding a new word after merging, leaving only one of the original two subwords, and eliminating the other subword (the case where two subwords appear consecutively at the same time).



```
1   #version: θ.2        14   q u
2   e n                  15   d e
3   o n                  16   i s
4   t i                  17   r 0
5   t h                  18   o r
6   e s</w>              19   a r
7   a n                  20   e s
8   o u                  21   m en
9   i n                  22   i t
10  r e                  23   i e </w>
11  e r                  24   s i
12  th e</w>             25   o m
13  o n</w>              26   a ti
```

Figure 4: Screenshot of the corresponding vocabulary based on the BPE decoding part.

(3) The number of words is reduced, which means that new words are generated after the vocabulary list is merged, and the original two subwords (two consecutive subwords that appear at the same time) are eliminated at the same time. In the actual experiment, as the number of combined words keeps increasing, the number of words in the combined vocabulary generally increases first and then decreases. Below we will introduce the specific coding of the subword model: after the above algorithm is used, the vocabulary of related subwords can generally be obtained, and the obtained vocabulary can be sorted according to the length relationship of the subwords. When encoding, for all input sequences, traverse the processed vocabulary to find out whether there is a token belonging to the input sequence. If present, it indicates that the character representation is one of the representations belonging to the input sequence. Generally traverse from the longest representation to the shortest representation, and try to use representations instead of substrings for each word. Finally, all representations are usually traversed and all substrings are replaced by vocabulary representations. The process of encoding is very computationally intensive. In practical applications, by premarking all words, the results of word representations are stored in the vocabulary. If there is an unknown word in the corpus in the vocabulary, the above coding strategy is used to mark the word, and then the new generated word representation is added to the vocabulary. Decoding is mainly done by splicing all the characters together and decoding through the above vocabulary. The following mainly shows the specific examples in the actual application process of this paper, as shown in Table 1.

3.2.2. Average Neural Network Structure. Transformer models can be fully parallelized during the training phase and can model source and target sentence inter- or intra-dependencies within short paths. The parallelization property of the model can quickly train neural machine translation, and the property of the interdependence of sequences contained in the attention mechanism gives transformer a powerful ability to

Table 1: Comparison of sentences processed by BPE.

English original	Mr Wohlfart may now reply to Mr McMahon's question, if he would like to do so
After processing	Mr Wo@@ h@@ l@@ far@@ *t* may now reply to Mr Mc@@ Ma@@ hon@@'s question, if he would like to do so
Original French	Nous espérons que la Commission traitera des points liés à l'additionnalité
After processing	Nous espérons que la Commission tra@@ itera des points liés à l'@@ additionn@@ alité

preserve sentence semantics and translation correlation alignment. However, because of the autoregressive generation mechanism in the decoder, the decoding of the transformer cannot enjoy the speed advantage of parallelization. And in the inference translation stage, the self-attention network in the model decoder even slows down the advantages it brings in the training stage. As can be intuitively seen in Figure 5, in the inference translation stage, in order to capture dependencies from previously predicted target words, self-attention in transformer needs to compute adaptive attention weights for all these words. However, RNN-based machine translation only needs the information of the previous one element, and CNN-based machine translation also only needs forward k elements (k represents the size of the convolution kernel). Therefore, how to maintain the training efficiency of transformer while accelerating its decoding efficiency has become a new and severe challenge.

In 2018, Zhang Biao et al. proposed the average attention network (AAN) to alleviate the above challenges. In this section, AAN is used to replace the self-attention computation part of the decoder part in the transformer. The average attention network structure is shown in Figure 5. Intuitively, it is equivalent to changing the dynamic attention weight calculation for each input in the previous self-attention mechanism into a simpler fixed weight (average), which indirectly saves the historical information of the previous position.

$$g_i = \mathrm{FFN}\left(\frac{1}{j}\sum_{k=1}^{j}\mathrm{input}_k\right). \tag{6}$$

Among them, FFN(•) is formula (6). Since it is an accumulation method, its input and output have the same dimension. In order to ensure the expressiveness of the model, after the cumulative average is used in this paper, it goes through a layer of linear full connection transformation. Due to the average accumulation method, the model cannot autonomously learn how much past information g_j should retain and how much new information should be captured from the current input input $_j$. Therefore, a gated network design is adopted to control the proportion of new and old information output through input gates and forgetting gates to increase the interpretability and stability of the model. The specific calculation formula is as follows:

$$\begin{aligned} i_j, f_j &= \sigma\left(w\left[\mathrm{input}_j; g_j\right]\right),\\ \tilde{h}_j &= i_j \odot \mathrm{input}_j + f_j \odot g_j,\\ h_j &= \mathrm{layer\,Norm}\left(\mathrm{input}_j + \tilde{h}_j\right). \end{aligned} \tag{7}$$

Among them, σ (.) is the sigmoid activation function, [. ; .] represents the direct splicing operation, which $\odot$ is the multiplication of the corresponding elements of the two matrices, and layerNorm (.) represents the output normalization of the sub-layer. The existence of a gating mechanism helps the model to detect direct correlations within the input embedding vectors. In order to stabilize the data distribution in the output and gradient descent, residual connections and normalization are directly used in the input layer and gated output layer.

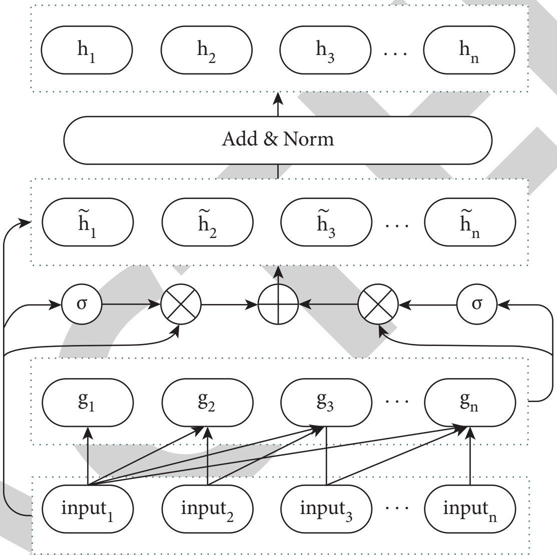


Figure 5: Schematic diagram of average attention network.

3.2.3. *Sparsity Normalization Function.* Sparse maximum has obvious advantages in some large-scale classification tasks, such as neural machine translation tasks, which require a softmax layer to establish a multinomial distribution of words for a huge vocabulary set. In the actual machine translation work, the researchers found that in the process of aligning the source sentence and the target sentence, the parts that should not be aligned are often aligned (the case of having a merit score). Second, in terms of attention mechanism, the focus of attention mechanism is the correlation calculation function, which calculates the relative importance of the input sequence and converts its data into a probabilistic representation. However, most of the time, the softmax computing mechanism used by the model will generate a relatively dense attention distribution. And because the output values of the softmax method are all positive numbers, the final decision is related to any sequence of the input. A commonly used softmax function is shown in (8), which is disadvantageous in applications that require sparse probability distributions, in which case a threshold is usually defined below which small probability values will be truncated directly to zero.

$$\mathrm{softmax}_i(x) = \frac{\exp(x_i)}{\sum_j \exp(x_j)}. \quad (8)$$

In the binary classification problem, the specific expression (9) of sparsemax is obtained by calculation and softmax is transformed into the sigmoid function in Chapter 2. As shown in Figure 6, it can be clearly seen that softmax and sparsemax are two-dimensional. Although sparsemax is a piecewise function, it asymptotically approximates the softmax function.

$$\mathrm{spare\,max}_1(Z) = \begin{cases} 1, & \text{if } t > 1, \\ \dfrac{t+1}{2}, & \text{if } -1 < t < 1, \\ 0, & \text{if } t < -1. \end{cases} \quad (9)$$

The above mainly introduces the basic theoretical basis of sparsemax, as well as the demonstration of its usability.

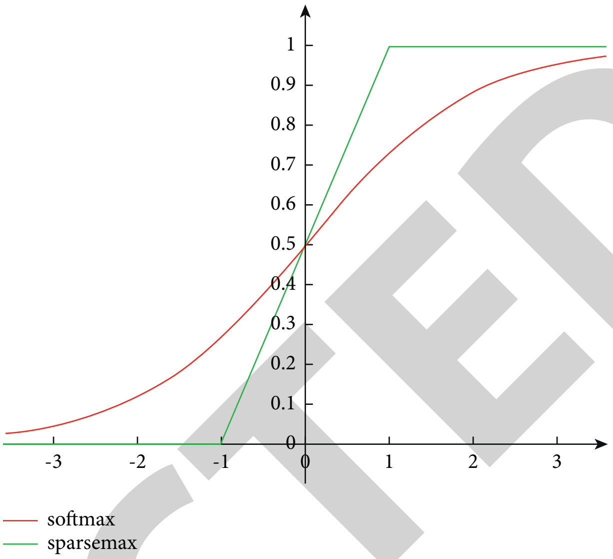


Figure 6: Schematic diagram of comparison of softmax and sparsemax in two-dimensional case.

4. Experiment and Result Analysis of Translation Model Based on Average Attention and Sparse Normalization

4.1. Experimental Data Preprocessing. Most of the original language text data have a lot of noise, so we must preprocess the data before it can be used by the model. Specific text processing should also use different methods for different corpora. For example, Chinese corpus does not have English-like spaces to separate words (English can directly use Python's split( ) function), so simple spaces and punctuation cannot be used to complete. In view of the special aspects of English, such as spelling problems, and the problem of missing or redundant letters in individual words, scholars generally use methods such as stem extraction and morphological restoration to analyze them. At present, researchers generally have a relatively complete corpus, such as wiki, for a specific task. This saves a lot of trouble for researchers to conduct scientific research. However, in many cases, some special fields need to be faced, and the existing corpus resources cannot meet the requirements of use. Therefore, it is necessary to use technical means such as crawlers to obtain data. In data cleaning, first of all, there are many html tags in most of the content obtained by the crawler, which need to be removed. Generally, Python's regex and other libraries are used to delete some nontext data, special characters, and punctuation not required by this corpus through regular expressions. Stop words, as the name suggests, have no special effect on people's understanding of the entire sentence after the word is removed. For example, the corpus generally contains many pronouns, function words, and some nouns and verbs that have no other meaning. These words usually have no effect on the researcher's model. Too much effect is more of a "burden," so stop words need to be removed. In English, most researchers can use the NLTK (Natural Language Toolkit) library and some specific packages under it. To deal with Chinese stop words, the relevant personnel usually need to face the task to construct Chinese stop words. There are more than 1200 regular Chinese stop words. After processing the above steps, it is necessary to standardize the input text to avoid the occurrence of training errors caused by word spelling and capitalization. In general, all characters of the text are converted into lowercase representations, as well as stem extraction (stemming, extracting word stems to unify the representation form) and part of speech restoration (lemmatization, converting all forms of language vocabulary into general forms); thus to help the model reduce the noise of the vocabulary, it is also more conducive to the final feature effect. In English corpus, spelling errors are very common, so spelling correction is particularly critical; usually, researchers use the Python-based open-source text processing TextBlob to process.

4.2. BLEU Evaluation Criteria. In 2002, IBM's Kishore et al. published an algorithm BLEU (Bilingual Evaluation Understudy) about machine translation bilingual evaluation indicators, which is currently the most popular low-cost and automated evaluation method, as shown in Table 2 1, the interpretation of the BLEU percentage score of the translation. The BLEU algorithm is also based on the n-gram model, and through a series of corrections, its results are more accurate. The original n-gram model treats consecutive n words in a sentence as a whole, and its accuracy is divided by the number of n-grams that exist simultaneously in the machine-translated translation and the reference translation (there can be multiple). Take the number of all n-grams in the machine translation. However, in such a calculation, there will be the same n-gram repeated in multiple reference translations at the same time, which will lead to the accumulation of the number of n-grams when solving, which is obviously not completely reasonable. In response to the above series of problems, Kishore et al. invented the

Table 2: Model comparison effect BLEU value.

Model	German-English	English-French
RNNsearch-50 + UNK	36.1	—
ConvS2S	25.2	40.5
Deep-Att + posUnk	—	39.2
GNMT + RL	24.6	39.9
Transformer	27.3	38.1
Transformer (big)	28.4	41.0
Ours	26.7	38.5

modified *n*-gram model to improve it. When the value of *n* continues to increase, the accuracy value of the modified n-gram decreases exponentially, so BLEU uses the modified *n*-gram for calculation through the logarithmic weighted average strategy; that is to say, the modified n-gram is added to the geometric mean. Generally, the maximum value of *n* is 4. However, doing so will result in shorter translations resulting in better scores, so Kishore et al. introduced a penalty factor to solve the problem of too short translations as is shown in Table 3.

The specific experimental results show that the geometric mean is more accurate than the arithmetic mean. The BLEU score of human translations is not necessarily close to 1. On the contrary, a good translation with high quality is closer to the reference translation. So for high-quality translations, the more the reference translations, the higher the BLEU score. Next, we mainly introduce the specific calculation and mathematical formula of BLEU score: first, the n-grams of each sentence are calculated in order, then, the number of n-grams of each sentence is accumulated, and then combined with the n-grams of all candidate sentences in the corresponding library divide the total number of grams so that the corrected precision np of all sentence banks can be solved, as shown in

$$p_n = \frac{\sum_{C\in\{\text{candidates}\}}\sum_{n-\text{gram}\in C}\text{count}_{\text{clip}}\,(n-\text{gram})}{\sum_{c_{,}\in\{\text{candidates}\}}\sum_{n-\text{gram}_{,}\in C}\text{count}_{\text{clip}}\,(n-\text{gram}')}. \tag{10}$$

4.3. Experimental Parameters, Settings, and Environment

4.3.1. Experimental Environment. TensorFlow developed by Google is an open-source Python external structure package for deep learning, which completes the internal operation by means of data flow graph. TensorFlow converts Python into more efficient C++ for code execution. TensorFlow greatly reduces the difficulty of getting started and developing deep learning. This section mainly introduces the specific experimental hardware and software environment of this article: (A) hardware environment—1. processor (CPU): Intel core i7, 2. memory (memory): 128 GB, 3. graphics card (GPU): GTX 1080 Ti, video memory size: 11 GB 4, hard disk: 2 TB; (B) software environment—1. computer language: Python 3.6, 2. operating system: Linux (Ubuntu 16.04), 3. CUDA: 9.0; cuDNN.

Table 3: BLEU score introduction.

BLEU (%) score	Interpretation of translation quality
<10	No real effect
10~19	Difficulty understanding key points
20~29	Key points clear, but with prominent syntax errors
30~40	Between understandable and good quality translations
40~50	Higher quality translation
50~60	High quality, fluent, and easy to understand
>60	Equivalent to high-quality human translation

4.3.2. Experimental Data and Experimental Parameter Design. Deep learning has been successfully used in commercial applications since the 1990s, but is generally an art that can only be performed by professionals rather than a general-purpose technique. This idea persisted until recent years because the development of the computer industry has weakened. The key reason for this is that training data are very precious in the past, and there are also large networks that tend to overfit and small networks that are difficult to perform well. Therefore, it must be trained through a strategy like regularization, which makes the training of the network as difficult as art rather than a practical technique. As society becomes more and more digitized, people's daily life trajectories appear on the Internet, and computers also record more and more daily data. Therefore, it is more convenient to organize data into electronic datasets suitable for deep learning algorithms, so the era of "big data" has promoted the progress of artificial intelligence and deep learning. The experimental data in this paper are mainly from public datasets. Some nonpublic fee-based datasets, which are expensive, do not provide much benefit for comparison and development in academic research. Therefore, the datasets in this paper mainly include the following types: the WMT (workshop on machine translation) dataset includes multiple public parallel corpora such as French-English, English-German, and Spanish-English. At the same time, a large number of previous researchers have also adopted this dataset. In order to make an objective and correct evaluation of the model in this paper, this paper also uses this dataset for experiments. Among them, the corpus shown in Table 4 is used in this paper.

The UN dataset, to aid the development of machine translation, contains parallel translation corpora between English, Chinese, Russian, French, Arabic, and Spanish. The dataset is the product of a compilation of UN documents in various languages over the years by Ziemski et al. Among them, there are about 25 million parallel sentence pairs in Chinese and French data. The dataset was processed by the relevant alignment algorithm when it was established, so the sentence structure was neat and formal. Among them, for the WMT German-English dataset, the training set contains about 2.0 M parallel sentences, and a validation set of more than 20,000 parallel sentences is set. The test set uses newtest2014, which has 3 k parallel sentences.

Table 4: Statistics of datasets in WMT.

Parallel datasets (L1-L2)	Number of sentences	L1 vocabulary	L2 vocabulary
German-English	1,920,209	44,548,491	47,818,827
French-English	2,007,723	51,388,643	50,196,035

Table 5: Hyperparameter design and specific meaning.

Parameter name	Numerical size	Specific meaning
batch_size	64	Minimum batch size
lr	0.0001	Learning rate
logdir	"-path" path	Path to save model parameters
maxlen	20	Maximum length of a sentence
min_cut	20	Minimum word frequency, use if it is less than
hidden_unit	512	Size of hidden layers and word embeddings
num_blocks	6	Number of stacked blocks to encode/decode
num_epoch	20	Rounds of iteration over the entire dataset
num_heads	8	Divide into several heads to calculate attention
dropout_rate	0.1	Drop-out rate
Sinusoid	False	Whether to use learned positional coding

When the model is trained, the vocabulary encoded by BPE is used, and the size is set to 40000. In addition to the WMT French-English dataset, the training set contains 2.0 M parallel sentences after removing the super-long sentences, the validation set has about 26 K sentence pairs, and the test set also uses newtest2014, including 3000 sentence pairs, which are also encoded by BPE. The vocabulary set is used to train the model, and the vocabulary size is set to 40000. In this paper, the Adam optimizer is used to optimize the parameters as $\beta_1 = 0.9$, $\beta 2 = 0.98$, and $\varepsilon = 10^{-9}$ The main hyperparameter designs of the experiments in this chapter are shown in Table 5.

The main hyperparameters of the experiments in this chapter are designed as shown in Table 5.

4.4. Experimental Results of This Paper. RNNsearch: The required intermediate hidden layer vector is generated by the encoder. The decoding process solves and outputs the context representation of the current moment through the attention mechanism and finally combines the decoding output and context representation of the previous moment to complete word prediction. CovS2S: This model is a neural translation model based on convolutional neural network proposed by Facebook. It saves the long-distance dependence of words in sentences through gating mechanism combined with multi-hop convolution and also makes the problem of gradient descent disappear during training. This model breaks the previous pattern of neural translation relying on recurrent neural networks and improves the parallel ability and model training efficiency during model training. It also has a very good effect on the translation effect.

Deep-Att + posUnk: This model adopts a deep LSTM network combined with a fast feedforward structure, passes through a fast network path without nonlinear transformation and recursive calculation, and does not need to pass gradients through a recurrent network. Gradient decay is much slower on this path, which makes the model easier to train and improves the performance of the model. GNMT + RL: This network proposes a corresponding solution for the high computational cost during deep learning training and translation and the rare words in the translated sentences. A low-precision algorithm is used, and the words are divided into common word units (word pieces).

The network model mostly uses RNN-based neural machine translation. In the actual training process, it cannot be trained in parallel, resulting in a lot of costs and training time, and the long-term dependencies in sentences are still not well resolved. In view of these model problems and advantages, the fusion strategy adopted in this chapter is mainly aimed at the transformer inference stage, and the decoding will become slower and slower with the length of the sentence. An average attention network is used to speed up the decoding process in the inference stage. In most translation models, the probability is calculated directly through softmax from the vector generated by the decoder to the vocabulary. In this chapter, sparsemax is used for the normalization calculation of the probability, and the unnecessary probability distribution is reduced in a sparse way. Make the generated related words have a high probability, and the rest are 0. Improve sparsity in predicting word probability distributions. In Table 6, compared with other models of the same type, the model proposed in this paper is significantly better than the translation models of RNN and CNN algorithms. Although transformer (big) has obvious advantages in BLEU score, transformer (big) model is about three times larger in size, has more parameters, and spends more training time. In summary, on the basis that transformer (base) is more efficient than the traditional RNN translation model (about 10 times faster), the translation speed of the entire model is further improved, and the actual effect is not much different. Therefore, the model has obvious advantages in efficiency and parameter scale. After experimenting with sparsemax on the average network, the actual effect is shown in Table 6. According to whether the feedforward neural network part in the average layer is used or not, relevant comparative experiments are designed. In the table, this paper conducts ablation experiments on the average attention layer, the gating mechanism, and the feedforward neural network layer, respectively. When the model only removes the feedforward network part of the AAN, it is found that the BLEU score of the model drops by about 0.3; when the model removes the gated part of the AAN, it is found that the BLEU score drops significantly by about 0.7. Therefore, it can be concluded that the gating mechanism in the AAN is more important to the translation effect of the model than the feedforward network. This paper believes that the main reason is that the gating mechanism can effectively control the flow of network information flow and ensure that the gradient descent is more stable during the training process. So, it has a higher importance in the network. The function of feedforward neural network in AAN can mainly capture richer input sequence information and increase the information expression ability of the module.

In addition, for the sparse normalization part, this paper conducts a comparative experiment between softmax and sparsemax. The results are shown in Table 7. Compared with the softmax, the BLEU score of the experiment is improved by about 0.2 points. Due to the softmax mechanism, the output of attention is not sparse enough. For tasks such as machine translation, most of them hope to obtain only words with high correlation with each translated word, but softmax assigns each word a certain amount of attention, resulting in insufficient attention. Therefore, this paper considers that it is only the result of inductive bias. So, using sparse attention can make the deviation from the ground truth weaker, and the experimental results also show this. Some researchers have also adopted the local attention mechanism to deal with such bias problems. However, in practice, discrete and nondifferentiable properties require Monte Carlo method for gradient approximation, which greatly increases the training complexity of the model. And sparsemax is differentiable, easy to calculate, and easy to use.

In this paper, the attention heat map after using sparsemax in the cross-attention layer of the decoder part is shown in Figure 7. It can be seen from the experimental results that after using sparse normalization to calculate the probability, the corresponding attention scores of some irrelevant words in Figure 7 are all zero (the color is white), which reduces the distribution of model data in the induction bias.. The accuracy of direct word alignment is improved, which not only improves the effect of the model, but also improves the interpretability of the model.

After processing some of the translation results of the fusion model, the results show that the overall translation effect is still fluent and the meaning of the sentence is relatively clear. Due to the existence of the attention mechanism, the effect of the word alignment of the model in this paper is still relatively neat.

Table 6: Comparative experiments of the model using the relevant sub-layers in the average layer.

Model	German-English
AAN	26.7
AAN (no FFN)	26.4
AAN (no gate)	25.9

Table 7: Comparative experiments of models using sparsemax.

Model	German-English
Softmax	26.5
Sparsemax	26.7

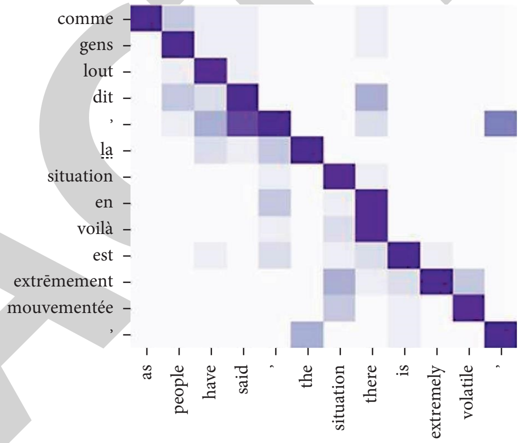


Figure 7: Matrix heatmap of attention scores in French-English translation.

5. Conclusion

With the rapid development of deep learning and society, people have more and more needs for the convenience brought by artificial intelligence. Neural machine translation plays a vital role in helping us communicate more quickly in our daily lives. With the development of deep learning, neural machine translation has also received attention and development by researchers. Especially in the application of encoder-decoder in natural language processing, the translation performance has been significantly improved. In 2014, the attention mechanism was used in neural machine translation, the performance of translation was greatly improved, and the interpretability of the model was increased. This research is based on the transformer machine translation model research experiments. Through the experimental results, we found that sparse normalization has a good effect on solving the inductive bias between too many data generated by softmax. Sparse normalization can not only improve the interpretability of the model, but also significantly improve the direct bilingual alignment in translation. This paper uses the AAN model to effectively solve the problem that the decoding of the transformer in the inference phase will waste computing resources as the sentence length becomes longer, and the solution can be achieved by caching the information contained in the previous moment. Experiments show that incorporating multi-representation word vector input can improve the translation effect of the model. In addition, this paper adopts the idea of scrutinizing neural network and integrates it into the transformer decoder part for the decoding speed of the transformer decoding part in the inference stage. The draft sentences generated by the first round of decoding are combined with the source sentences, so that more abundant overall sentence information can be obtained in the deliberation and decoding stage, which improves the fluency and integrity of translation. It makes our understanding of neural machine translation and the nature of attention mechanism more profound.

Data Availability

The dataset can be accessed upon request.

Conflicts of Interest

The author declares that there are no conflicts of interest.

References

[1] P. Zheng, "Multisensor feature fusion-based model for business English translation," *Scientific Programming*, vol. 2022, Article ID 3102337, 10 pages, 2022.

[2] H. Mi, B. Sankaran, Z. Wang, and A. Ittycheriah, "Coverage embedding models for neural machine translation," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, Texas USA, November 2016.

[3] F. Xia, L. Wu, Y. Tao, and X. Liu, "Deliberation networks: sequence generation beyond one-pass decoding," *News in Physiological Sciences*, vol. 30, 2017.

[4] A. Radford, J Wu, and R Child, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, 2019.

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

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate[J]," *Computer ence*, 2014.

[7] M. T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," *Computer Science*, 2015.

[8] Y. Wu, M. Schuster, Z. Chen, and V. Le Quoc, "Google's neural machine translation system: bridging the gap between human and machine translation," 2016, https://arxiv.org/abs/1609.08144.

[9] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "Convolutional sequence to sequence learning," *Proceedings of the 34th International Conference on Machine Learning*, vol. 70, 2017.

[10] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," 2016, https://arxiv.org/abs/1611.02344.

[11] A. Vaswani, N. Shazeer, N. Parmar, and J. Uszkoreit, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[12] G. Lample, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," 2017.

[13] T. Mikolov, K. Chen, G. Corrado, and D. Jeffrey, "Efficient estimation of word representations in vector space," 2013, https://arxiv.org/abs/1301.3781.

[14] G. Klein, Y. Kim, Y. Deng, and M. R. Alexander, "OpenNMT: open-source toolkit for neural machine translation," 2017, https://arxiv.org/abs/1701.02810.

[15] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," 2012, https://arxiv.org/abs/1212.5701.

[16] J. Chung, K. Cho, and Y. Bengio, "A character-level decoder without explicit segmentation for neural machine translation," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2016.

[17] S. Sukhbaatar, A. Szlam, J. Weston, and R Fergus, "End-to-end memory networks," *Computer Science*, vol. 28, 2015.

[18] N. Kalchbrenner, L. Espeholt, K. Simonyan, and O Aaron van den, "Neural machine translation in linear time," 2016, https://arxiv.org/abs/1610.10099.

[19] Z. Tu, Z. Lu, Y. Liu, and X Liu, "modeling coverage for neural machine translation," 2016, https://arxiv.org/abs/1601.04811.

[20] T Gongbo, M. Müller, A. Sennrich, and S Rico, "Why self-attention? A targeted evaluation of neural machine translation architectures," 2018, https://arxiv.org/abs/1808.08946.

[21] H. Q. Nguyen, T. M. Nguyen, H. H. Vu, and V. V Nguyen, "An effective coverage approach for attention-based neural machine translation," in *Proceedings of the 2019 6th NAFOTED Conference on Information and Computer Science (NICS)*, IEEE, Hanoi, Vietnam, December 2020.

[22] N. T. Tran, V. Luong, N. L. T. Nguyen, and M. Q Nghiem, "Effective attention-based neural architectures for sentence compression with bidirectional long short-term memory," in *Proceedings of the 7th Symposium on Information and Communication Technology*, Ho Chi Minh City, Vietnam, December 2016.

[23] N. Kalchbrenner and P. Blunsom, "Recurrent convolutional neural networks for discourse compositionality," 2013, https://arxiv.org/abs/1306.3584.