International Journal for Housing Science and Its Applications
Publish August 6, 2025. Volume 46, Issue 3 Pages 6835-6848

https://doi.org/10.70517/ijhsa463586

Using Neural Network Language Modeling to Improve the English Translation of Specialized Terms in Computer Translation

Yuan Wang[1,*], Jing Jin[1], Meiling Ye[2] and Tingting Tao[2]
[1] School of International Studies, Maanshan Teacher's College, Maanshan, Anhui, 243041, China
[2] Department of General Education, Maanshan Teacher's College, Maanshan, Anhui, 243041, China
Corresponding authors: (e-mail: m17755502925@163.com).

Abstract Machine translation technology has been developing rapidly in general-purpose fields, but accurate translation of specialized terminology is still difficult. In some specialized fields, the accuracy of terminology translation has a significant impact on the overall translation quality, and specialized improvement methods for terminology translation need to be developed. In this paper, we propose an English translation neural network model that incorporates terminology information with the aim of improving the English translation of specialized terminology in computer translation. Taking the Transformer model with self-attention mechanism as the baseline, the model is optimized through three key steps: firstly, the bilingual corpus is customized with terminology dictionaries for terminology segmentation; secondly, the word vectors are trained on the general corpus and the specialized corpus by combining two methods, Glove and Word2vec, respectively, and are used for the initialization of parameters in model embedding layer; and lastly, for the problem of unregistered words, the introduction of the external terminology dictionary for lookup replacement. The experimental results show that on the electrical engineering domain corpus, the BLEU value of the proposed model reaches 35.1%, which is 1.36% higher than that of the baseline model; during the training process, the performance advantage of the present model continues to be stable when more than 10,000 steps are taken; compared with the other seven translation models, the present model obtains the optimal translation effect by increasing the training time by only 5.86%. The experiment proves that the English translation neural network model fusing terminology information can effectively improve the translation quality of specialized terminology, and provides a new solution for machine translation in vertical domains.

Index Terms neural network language model, machine translation, self-attention mechanism, specialized terminology translation, word vector pre-training, Transformer model

I. Introduction

In the context of globalization and informationization, English, as an international common language, plays an irreplaceable role in various fields. In the translation of English terminology, there exists a double test of professionalism and accuracy, and its accurate translation is not only the conversion of the language level, but also the bridge of professional knowledge and cultural exchange [1]. On the one hand, English terminology originates from its profound disciplinary background and professional knowledge, which may produce misunderstanding or misinterpretation once translated inaccurately, thus affecting the effect of academic communication and cooperation [2], [3]. This requires that the accurate translation of professional terms not only needs to maintain the meaning of the original text, but also needs to conform to the expression habits of the target language to ensure that readers can understand correctly [4], [5]. On the other hand, the translation of English terminology is not only a technical task, but also a deep cultural dialog [6].

Since English terminology is often rooted in specific cultural backgrounds and contexts, its meanings and usages may differ significantly among different cultures [7], [8]. This requires that the translation of English terminology not only accurately conveys the original meaning, but also conforms to the cultural background and expression habits of the target language, in order to adapt to the needs of readers in different cultural contexts [9]-[11]. With the advancement of artificial intelligence and machine learning technology, automatic translation tools have been popularized, so it is particularly important to study in depth the translation strategies of English terminology in computer translation to improve the quality of machine translation [12], [13].

Machine translation technology has gone through the development process of rule-based machine translation, statistical-based machine translation to the current neural network machine translation. In recent years, neural network-based machine translation models have achieved remarkable results in general domain translation tasks, especially the Transformer model has become the mainstream architecture by virtue of its parallel processing capability and its ability to capture long-distance dependencies. However, in specialized domain translation tasks, due to the complexity and specificity of specialized terms, existing translation models are often difficult to accurately convey the specialized connotations of terms, resulting in poor translation quality. Specialized terminology is the core component of texts in specialized fields, and its translation accuracy directly affects the overall translation quality. Traditional neural machine translation models face two major challenges when dealing with specialized terms: on the one hand, the relative scarcity of bilingual parallel corpus in specialized domains makes it difficult to support the model to fully learn the translation laws of the terms; on the other hand, many specialized terms belong to the low-frequency words or unregistered words, which are easy to be incorrectly processed in the translation process. These problems have led to unsatisfactory results of machine translation in specialized fields, restricting the application of machine translation technology in specialized fields. As a typical specialized field, electrical engineering contains a large number of terms in its text, which requires high translation accuracy and provides an ideal scenario for specialized term translation research. Existing studies have shown that integrating external knowledge into neural machine translation models can effectively improve translation quality, especially for specialized terms. However, how to effectively fuse terminological information while balancing model complexity and translation effect is still an issue that needs to be explored in depth.

This study proposes an English translation neural network model for fusing terminology information, which is optimally designed based on the Transformer architecture. The study includes three main aspects: first, making full use of the existing terminology dictionary, introducing customized word segmentation methods in the data preprocessing stage to improve the term recognition accuracy; second, innovatively combining two word vector pre-training methods, Glove and Word2vec, extracting term representations from general and specialized corpus respectively, and optimizing the initialization of the model's embedding layer parameters; and finally, designing an unregistered word Find and Replace mechanism, which utilizes external dictionaries to accurately replace the unregistered terminology words that appear in the translation process. The goal of this multi-level terminology information fusion strategy is to maximize the translation quality of professional terms while keeping the model structure simple and efficient, providing a new solution for machine translation in vertical domains.

II. Transformer model based on self-attention mechanism

Machine translation is a technology that utilizes computers to convert the source language into the target language, which is highly likely to produce low-quality translations compared to human beings. Therefore, the neural network language model is utilized to improve computer translation, and the Transformer neural network model based on the attention mechanism is used as the baseline model, and the in-domain terminology dictionary and monolingual corpus are utilized to fuse the terminology information, so as to propose an English translation model that fuses terminology information to enhance the English translation of specialized terminology. The Transformer model based on self-attention mechanism is as follows.

Self-attention mechanism is adopted in the translation process of Transformer model, and neural network structures such as RNN and CNN are abandoned to improve the translation efficiency and also obtain accurate translation results. The encoder of Transformer consists of n identical encoder layers, and each encoder layer includes a multi-head attention sublayer and a fully-connected feed-forward neural network sublayer. In order to mitigate the gradient vanishing problem, residual connections are introduced between every two sublayers, followed by layer regularization operations, which are designed to make gradient transfer easier and speed up model convergence, see equation (1):

$$layernorm\left(h+sublayer(h)\right) \tag{1}$$

where, $layernorm(\cdot)$ denotes the output of the layer regularization function and $sublayer(\cdot)$ denotes the sublayer output. h denotes the hidden state of the sublayer input.

The decoder consists of a stack of n identical decoder layers, and in contrast to the encoder, the decoder has an additional encoding-decoding cross-attention sublayer to combine the information from the encoder and the decoder, again requiring the use of residuals to connect between each of the two sublayers, followed by a layer regularization operation.The basic unit of the multi-head attention mechanism in the encoder sublayer and the decoder sublayer of the Transformer is the scaled dot product attention mechanism, whose inputs are the query matrix Q, the key-value pair matrix K and V. The dot product attention mechanism training process is as follows.

The source sentence, after word embedding and positional embedding, yields the input sequence $X=(x_1,x_2,\ldots,x_n)\in R^{d_x}$. Define three matrices W_Q, W_K, W_V, and use W_Q, W_K, and W_V to perform a linear transformation operation on the input sequences, respectively, which in turn produces three new vectors, q_t, k_t, and v_t. Put all q_t into one large matrix, denoted as the query matrix Q, all k_t into one large matrix, denoted as the key matrix K, and all v_t into one large matrix, denoted as the value matrix V. The attention weight of the first word is obtained by multiplying the query vector q_1 of the first word with the key matrix K. After that, it is also necessary to pass the obtained values through $soft\max$ so that they sum to 1. The obtained weights are multiplied by the value vectors v_t of the corresponding words respectively in a weighted summation to obtain the output of the first word. Continuing the above operation for subsequent input vectors yields all outputs after passing through the dot product attention mechanism, see equation (2):

$$Attention(Q,K,V)=soft\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

Finally the multi-head attention mechanism splices the results of multiple dot product attention mechanisms to obtain the final output. See equation (3), equation (4):

$$head_i=Attention\left(QW^Q,KW^K,VW^V\right) \tag{3}$$

$$MultiHead(Q,K,V)=Concat\left(head_i\right)W^O \tag{4}$$

where, W^Q,W^K,W^V,W^O is the parameter matrix. $head_i$ is the output vector of the i th attention head.

The feed-forward neural network is a fully connected network layer between two layers, and RELU is used as the activation function, see equation (5):

$$FFN(x)=\max\left(0,xW_1+b_1\right)W_2+b_2 \tag{5}$$

where, W_1,W_2,b_1,b_2 is the model parameter.

Since Transformer does not contain recurrent and convolutional neural networks that process each word in a sentence in parallel and does not have the ability to recognize the order of each word, the addition of positional encoding can help the translation model to recognize the positional information of the words in a sentence. The dimension of positional encoding is the same as that of word vectors, and the result of adding the two can be used as the input sequence.Transformer calculates positional encoding by using sine and cosine functions with different frequencies, see Eq. (6), Eq. (7):

$$PE_{(pos,2i)}=\sin\left(pos/10000^{2i/d_{model}}\right) \tag{6}$$

$$PE_{(pos,2i+1)}=\cos\left(pos/10000^{2i/d_{model}}\right) \tag{7}$$

where, pos is the absolute position of the element in the sequence, $2i$ is the current dimension of the position-encoded vector, and d_{model} denotes the dimension of the input sequence.

III. A Neural Network Model for English Translation Incorporating Terminological Information

First, the bilingual corpus is customized with terminology dictionaries during data preprocessing, and word vectors containing generic text information and terminology information are obtained by training with large-scale monolingual corpus through Glove and Word2vec, respectively. Secondly, the word vectors obtained by combining the two training initialize the embedding layer parameters of the model. Finally, for the problem of unregistered words due to in-domain terminology in the translation process, an external terminology dictionary is used to find and replace them.

III. A. 3.1 Word vector pre-training

In natural language processing, the representation of word vectors has evolved from the original one-hot encoding to the later distributed representation, which compresses the originally sparse huge dimensions into a space of smaller dimensions and makes the word representation contain more semantic information. In this paper, we use two pre-trained word vector methods: glove and Word2vec.

III. A. 1) 3.1.1 Glove model

The Glove model is a word vector generation model based on global word frequency statistics. There are two main types of word vector generation methods: (1) word vector generation methods based on word co-occurrence matrices, such as latent semantic analysis (LSA). These methods can effectively utilize the global statistical information of the document set and have faster training speed, but the disadvantage is that the most frequently occurring words may convey relatively little semantic relevance and contribute disproportionately to the similarity measure. (2) Window-based word vector generation methods, such as word2vec. Such methods are able to capture complex syntactic information including word similarity and learn word-local contextual features, but cannot fully utilize the statistical information. The Glove model combines LSA and word2vec by traversing the document set using a sliding window and updating the word co-occurrence matrix within the window. It also integrates the advantages of LSA to construct global word co-occurrence matrix and word2vec window mechanism. The Glove model training process is shown below:

(1) Creation of "word-word" level co-occurrence matrices X

Set a window of fixed size and slide from left to right in the document set, the word located in the center of the window is called the center word, and the words located on both sides of the center word in the window are called the context. X_{ij} denotes the number of times the word j occurs when the word i is the center word. $X_i = \sum_k X_{i,k}$ denotes the total number of times the word i is the center word. $P_{ij} = P(j \mid i) = \frac{X_{ij}}{X_i}$ denotes the probability of occurrence of word j when word i is the center word. The co-occurrence matrix can be obtained by continuously sliding the window and counting the total number of times the word in the corresponding position co-occurs with its context word when it is the center word.

(2) When dealing with the co-occurrence matrix X, a batch of non-zero items are firstly extracted as training samples, and the corresponding word embeddings and their bias terms are randomly initialized for the selected samples.

(3) Perform dot product and offset operations to compute the deviation from the target value, and thus obtain the loss value with respect to $\log\left(X_{ik}\right)$.

(4) Finally, the gradient descent method is applied to adjust the word embeddings and bias terms through the backpropagation algorithm, iterating the process until the value of the loss function is minimized, as indicated by Equation (8).

Using the ratio of probabilities is better for finding word-to-word relationships compared to using probabilities directly. Therefore, the loss function J is constructed so that the function $F\left(w_i, w_j, w_k\right) = \frac{p_{ik}}{p_{jk}}$ holds for the calculation of the ratio of the word vectors where w_i, w_j, w_k are the word vectors of the words i, j, and k, respectively. The purpose of training Glove is to minimize the loss function, i.e., to make the inner product of word vectors plus bias converge to the logarithm of the number of co-occurrences as much as possible. The final optimized loss function is given using a weighted least squares regression model:

$$J = \sum_{i,j=1}^{N} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log\left(X_{ij}\right)\right)^2 \tag{8}$$

where w_i, $\tilde{w}_j$ denote the word vectors of the words i, j, respectively, b_i, $\tilde{b}_j$ correspond to the word vectors w_i, $\tilde{w}_j$ bias, $\log\left(X_{ij}\right)$ is the logarithm of the number of co-occurrences of word pairs formed by words i, j, and N denotes the number of words in the lexicon in the document set. $f\left(X_{ij}\right)$ is the weight function to control the effect of different sizes of co-occurrence counts X_{ij} on the result, and the formula is shown in equation (9):

$$f(x) = \begin{cases} (x / x_{\max})^{\partial} & if\ x < x_{\max} \\ 1 & otherwise \end{cases} \tag{9}$$

Equation (9) shows that the more times a word pair co-occurs, the more that word pair has a greater weight to be penalized, and that word pairs with fewer co-occurrences have a relatively smaller weight, allowing word pairs that do not often co-occur within the same window to provide some degree of influence on the results without causing the results to be unduly skewed in favor of word pairs that co-occur a great deal more often.

III. A. 2) 3.1.2 Word2vec modeling

Word2vec model is a distributed representation based on neural networks, i.e., words or phrases are mapped by a simple neural network into a real vector, and the length of each real vector can be artificially specified, so that the resulting word vectors can well express the semantic relationship between words while implying the original meaning.Word2vec model contains two kinds: the Skip-Gram model and the Continuous Bag of Words model (CBOW). The model needs to adjust the weights with a large amount of data in order to prevent the occurrence of overfitting phenomenon during the training process, then the time complexity of the model will be high. The two efficient training methods of layer order Softmax and negative sampling (NEG) are needed to optimize the training process of the model.

(1) Skip-byte model

The jump byte model is a simple three-layer neural network structure. The model is characterized by the prediction of its context words w_k, w_{k-1}, w_{k+1} and w_{k+2} by the center word w_k.

The input layer is the center word w, which is expressed as w_k in One-Hot coding, and there exists a weight matrix W from the input layer to the hidden layer, and the hidden layer h is the product of the weight matrix W and the input layer, and there exists a weight matrix W' from the hidden layer to the output layer, and each output value of the output layer is a vector of the hidden layer h dot-multiplied by the weight matrix W'.

(2) CBOW model

The CBOW model is a three-layer neural network structure that features inputs of known context words w_{k-2}, w_{k-1}, w_{k+1}, and w_{k+2} outputs of predictions for the central word w_k.

The CBOW model is a reversal of the hopping byte model, and the principle is similar to the hopping byte model. First, the weight matrix of the hidden layer is randomly initialized, and then the context word is used to predict the center word, the model will constantly correct the weight matrix of the hidden layer during the training process, and finally the final result w_k can be obtained by using the product of the vector of the hidden layer and the weight matrix.

(3) Layer Order Softmax

The layer order Softmax-based model is mainly composed of a three-layer neural network, including input layer, hidden layer and output layer. In the Word2vec method, the objective function to be optimized based on the skip-byte model under this method is:

$$L = \sum_{w \in C} \log p\left(Context(w) \mid w\right) \tag{10}$$

The objective function to be optimized based on the CBOW model under this method is:

$$L = \sum_{w \in C} \log p\left(w \mid Context(w)\right) \tag{11}$$

where C denotes the corpus, w denotes any word in C, and $Context(w)$ denotes the context word of w.

1) Hierarchical Softmax-based Skip Byte Model The efficiency of the hierarchical Softmax method is mainly reflected in the output layer, which is a Huffman tree structure.

The Huffman tree is categorized in such a way that the positive class is denoted as 0 and the negative class is denoted as 1. The Sigmoid function is used to determine the probability of a node being classified into a positive class as:

$$\sigma\left(x_w^T \theta\right) = \frac{1}{1 + e^{-x_w^T \theta}} \tag{12}$$

The probability of being classified into a negative category is:

$$1 - \sigma\left(x_w^T \theta\right) = 1 - \frac{1}{1 + e^{-x_w^T \theta}} \tag{13}$$

In the jump-byte model, the conditional probability function $p\left(Context(w) \mid w\right)$ can be generally defined as:

$$\begin{aligned} p\left(Context(w) \mid w\right) &= \prod_{u \in Context(w)} p\left(u \mid w\right) \\ &= \prod_{u \in Context(w)} \prod_{j=2}^{l^u} p\left(d_j^u \mid v(w), \theta_{j-1}^u\right) \end{aligned} \tag{14}$$

Among them:

$$p\left(d_j^u \mid v(w), \theta_{j-1}^u\right)=\begin{cases}\sigma\left(v(w)^T \theta_{j-1}^u\right) & d_j^u=0 \\ 1-\sigma\left(v(w)^T \theta_{j-1}^u\right) & d_j^u=1\end{cases} \tag{15}$$

Substituting Eq. (14) into Eq. (10) gives:

$$L=\sum_{w \in C} \log \prod_{u \in Context(w)} \prod_{j=2}^{l^u}\left[\sigma\left(v(w)^T \theta_{j-1}^u\right)\right]^{1-d_j^u} \cdot\left[1-\sigma\left(v(w)^T \theta_{j-1}^u\right)\right]^{d_j^u} \tag{16}$$

The objective function of the jump byte model can be obtained by organizing Eq. (16):

$$L=\sum_{w \in C} \sum_{u \in Context(w)} \sum_{j=2}^{l^u} Q_1 \tag{17}$$

Among them:

$$Q_1=\left(1-d_j^u\right) \cdot \log\left[\sigma\left(v(w)^T \theta_{j-1}^u\right)\right]+d_j^u \log\left[1-\sigma\left(v(w)^T \theta_{j-1}^u\right)\right] \tag{18}$$

The parameters are optimized using the gradient ascent method and the partial derivatives of Eq. (17) with respect to θ_{j-1}^u:

$$\frac{\partial L}{\partial \theta_{j-1}^u}=\left[1-d_j^u-\sigma\left(v(w)^T \theta_{j-1}^u\right)\right] \cdot v(w) \tag{19}$$

Then the update formula for θ_{j-1}^u is:

$$\theta_{j-1}^u=\theta_{j-1}^u+\lambda\left[1-d_j^u-\sigma\left(v(w)^T \theta_{j-1}^u\right)\right] \cdot v(w) \tag{20}$$

Similarly about the partial derivatives of $v(w)$:

$$\frac{\partial L}{\partial v(w)}=\left[1-d_j^u-\sigma\left(v(w)^T \theta_{j-1}^u\right)\right] \cdot \theta_{j-1}^u \tag{21}$$

Then the update formula for $v(w)$ is:

$$v(w)=v(w)+\lambda \sum_{u \in Context(w)} \sum_{j=2}^{l^u}\left\{\left[1-d_j^u-\sigma\left(v(w)^T \theta_{j-1}^u\right)\right] \cdot \theta_{j-1}^u\right\} \tag{22}$$

2) CBOW model based on layer order Softmax

In the CBOW model, its conditional probability function $p\left(w \mid Context(w)\right)$ can be generally defined as:

$$p\left(w \mid Context(w)\right)=\prod_{j=2}^{l^w} p\left(d_j^w \mid x_w, \theta_{j-1}^w\right) \tag{23}$$

Among them:

$$p\left(d_j^w \mid x_w, \theta_{j-1}^w\right)=\begin{cases}\sigma\left(x_w^T \theta_{j-1}^w\right) & d_j^w=0 \\ 1-\sigma\left(x_w^T \theta_{j-1}^w\right) & d_j^w=1\end{cases} \tag{24}$$

Substituting Eq. (23) into Eq. (11) gives:

$$L=\sum_{w \in C} \log \prod_{j=2}^{l^w}\left[\sigma\left(x_w^T \theta_{j-1}^w\right)\right]^{1-d_j^w} \cdot\left[1-\sigma\left(x_w^T \theta_{j-1}^w\right)\right]^{d_j^w} \tag{25}$$

The objective function of the CBOW model can be obtained by organizing Eq. (25):

$$L=\sum_{w\in C}\sum_{j=2}^{l^w}Q_2 \tag{26}$$

Among them:

$$Q_2=\left(1-d_j^w\right)\cdot\log\left[\sigma\left(x_w^T\theta_{j-1}^w\right)\right]+d_j^w\cdot\log\left[1-\sigma\left(x_w^T\theta_{j-1}^w\right)\right] \tag{27}$$

To maximize the objective function of the CBOW model, the parameters are optimized using the gradient ascent method, and the partial derivatives of Eq. (26) with respect to θ_{j-1}^w:

$$\frac{\partial L}{\partial\theta_{j-1}^w}=\left[1-d_j^w-\sigma\left(x_w^T\theta_{j-1}^w\right)\right]\cdot x_w \tag{28}$$

Then the update formula for θ_{j-1}^w is:

$$\theta_{j-1}^w=\theta_{j-1}^w+\lambda\left[1-d_j^w-\sigma\left(x_w^T\theta_{j-1}^w\right)\right]\cdot x_w \tag{29}$$

Similarly about the partial derivatives of x_w:

$$\frac{\partial L}{\partial x_w}=\left[1-d_j^w-\sigma\left(x_w^T\theta_{j-1}^w\right)\right]\cdot\theta_{j-1}^w \tag{30}$$

Then the update formula for $v(w')$ is:

$$v(w')=v(w')+\lambda\sum_{j=2}^{l^w}\left[1-d_j^w-\sigma\left(x_w^T\theta_{j-1}^w\right)\right]\cdot\theta_{j-1}^w \tag{31}$$

where $w'\in Context(w)$.

(4) Negative Sampling

The essence of negative sampling is to use the known probability density function to estimate the unknown probability density function, the purpose is to improve the speed of training and the quality of transformed word vectors. Negative sampling is different from the stratified Softmax method in that negative sampling no longer uses the Huffman tree structure, but a simple random negative sampling method, which can substantially improve the training performance.

For the corpus C, each word in the dictionary has a high or low frequency of occurrence, and for words with a higher frequency of occurrence, the probability of being selected as a negative sample should be larger, and vice versa, the probability of being selected should be smaller, which is a problem of weighted sampling.

Let any word w in the dictionary D correspond to a length $l(w)$, Eq:

$$l(w)=\frac{Counter(w)}{\sum_{u\in C}Counter(u)} \tag{32}$$

Where $Counter(w)$ denotes the frequency of occurrence of the word w in the corpus.

For the sampling part, all the $l(w)$ are first connected to form a unit line segment of length “1”. If a prime is randomly thrown onto this line segment, the longer the length, the higher the probability of being hit, which establishes a mapping relationship.

1) Negative Sampling Based Skip Byte Modeling

For $\forall w'\in D$, define the label of the word w':

$$L^w(w')=\begin{cases}1 & w'=w\\ 0 & w'\neq w\end{cases} \tag{33}$$

where positive samples are 1 and negative samples are 0.

First a subset of negative samples $NEG^{w'}(w)\neq\varnothing$ of a w' is selected, and for a given sample $(w,Context(w))$, the objective function is:

$$g(w)=\prod_{w'\in Context(w)}\prod_{u\in w\cup NEG^{w'}(w)}p(u\mid w') \tag{34}$$

Among them:

$$p(u \mid w') = \begin{cases} \sigma\left(v(w')^T \theta^u\right) & L^w(u) = 1 \\ 1 - \sigma\left(v(w')^T \theta^u\right) & L^w(u) = 0 \end{cases} \tag{35}$$

Collating Eq. (34) gives:

$$g(w) = \prod_{w' \in Context(w)} \prod_{u \in w \cup NEG^{w'}(w)} \left\{ \left[\sigma(v(w')^T \theta^u)\right]^{L^w(u)} \cdot \left[1 - \sigma(v(w')^T \theta^u)\right]^{1 - L^w(u)} \right\} \tag{36}$$

Given a corpus \(C\), the overall optimization objective is:

$$G = \prod_{w \in C} g(w) \tag{37}$$

Taking the logarithmic operation of Eq. (37) yields the final objective function:

$$L = \sum_{w \in C} \log g(w) = \sum_{w \in C} \sum_{\tilde{w} \in Context(w)} \sum_{u \in w \cup NEG^{w'}(w)} Q_3 \tag{38}$$

Among them:

$$Q_3 = L^w(u) \log\left[\sigma(v(w')^T \theta^u)\right] + [1 - L^w(u)] \log\left[1 - \sigma(v(w')^T \theta^u)\right] \tag{39}$$

The parameters are optimized using the gradient ascent method and the partial derivatives of Eq. (38) with respect to θ^u:

$$\frac{\partial L}{\partial \theta^u} = \left[L^w(u) - \sigma(v(w')^T \theta^u)\right] \cdot v(w') \tag{40}$$

Then the update formula for θ^u is:

$$\theta^u = \theta^u + \lambda \left[L^w(u) - \sigma(v(w')^T \theta^u)\right] \cdot v(w') \tag{41}$$

Similarly the partial derivative of $v(w')$ can be obtained:

$$\frac{\partial L}{\partial v(w')} = \left[L^w(u) - \sigma(v(w')^T \theta^u)\right] \cdot \theta^u \tag{42}$$

Then the update formula for $v(w')$ is:

$$v(w') = v(w') + \lambda \sum_{u \in w \cup NEG^{w'}(w)} \left[L^w(u) - \sigma(v(w')^T \theta^u)\right] \cdot \theta^u \tag{43}$$

where $w' \in Context(w)$.

2) CBOW model based on negative sampling

First a negative sample set $NEG(w) \neq \varnothing$ of $Context(w)$ is selected, and for a given sample $(Context(w), w)$, the objective function is:

$$g(w) = \prod_{u \in w \cup NEG(w)} p(u \mid Context(w)) \tag{44}$$

Among them:

$$p\left(u \mid Context(w)\right) = \begin{cases} \sigma\left(x_w^T \theta^u\right) & L^w(u) = 1 \\ 1 - \sigma\left(x_w^T \theta^u\right) & L^w(u) = 0 \end{cases} \tag{45}$$

Collating Eq. (44) gives:

$$g(w)=\prod_{u\in NEG(w)}\left[\sigma\left(x_w^T\theta^u\right)\right]^{L^w(u)}\cdot\left[1-\sigma\left(x_w^T\theta^u\right)\right]^{1-L^w(u)} \tag{46}$$

where x_w is the sum of word vectors for each word in $Context(w)$. Increasing the probability of positive samples while decreasing the probability of negative samples, the overall optimization objective for a given corpus C is thus:

$$G=\prod_{w\in C}g(w) \tag{47}$$

Taking logarithmic operations on Eq. (47) yields the final objective function:

$$L=\log G=\sum_{w\in C}\log g(w) \tag{48}$$

Substituting Eq. (46) into Eq. (48) yields:

$$\begin{aligned}L&=\sum_{w\in C}\sum_{u\in w\cup NEG(w)}\log\left[\sigma(x_w^T\theta^u)\right]^{L^w(u)}\cdot\left[1-\sigma(x_w^T\theta^u)\right]^{1-L^w(u)}\\&=\sum_{w\in C}\sum_{u\in w\cup NEG(w)}\left\{L^w(u)\cdot\log\sigma(x_w^T\theta^u)+[1-L^w(u)]\cdot\log\left[1-\sigma(x_w^T\theta^u)\right]\right\}\end{aligned} \tag{49}$$

In the following, the parameters are optimized using the gradient ascent method and the partial derivatives of Eq. (49) with respect to θ^u:

$$\begin{aligned}\frac{\partial L}{\partial\theta^u}&=\frac{\partial}{\partial\theta^u}\left\{L^w(u)\cdot\log\sigma(x_w^T\theta^u)+[1-L^w(u)]\cdot\log\left[1-\sigma(x_w^T\theta^u)\right]\right\}\\&=\left[L^w(w)-\sigma(x_w^T\theta^u)\right]\cdot x_w\end{aligned} \tag{50}$$

Then the update formula for θ^u is:

$$\theta^u=\theta^u+\lambda\left[L^w(w)-\sigma(x_w^T\theta^u)\right]\cdot x_w \tag{51}$$

The partial derivatives of x_w are obtained in the same way:

$$\frac{\partial L}{\partial x_w}=\left[L^w(u)-\sigma(x_w^T\theta^u)\right]\cdot\theta^u \tag{52}$$

Then the update formula for $v(w)$ is:

$$v(w)=v(w)+\lambda\sum_{u\in w\cup NEG(w)}\left[L^w(u)-\sigma(x_w^T\theta^u)\right]\cdot\theta^u \tag{53}$$

In this paper, word vector representations containing more terminological information are obtained by Word2vec training using a large-scale specialized domain monolingual corpus, combined with word vector representations obtained by Glove training on a large-scale generalized monolingual corpus, so that each word in the corpus is adequately represented by a vector.

III. B. 3.2 Initialization of Embedding Layer Parameters

Currently, research on machine translation can be categorized into general-purpose domain machine translation and vertical-specific domain machine translation. The former uses a general-purpose corpus for training, and its trained neural machine translation model translates excellently in general semantic environments and performs poorly in domain-specific specialized texts. The latter mostly use only a limited bilingual corpus in their domain as the training set, which restricts the word vectors from learning the information contained in the words themselves, resulting in weak generalization ability of the trained models. In this paper, we use Word2vec to train word vectors containing terminology information and combine them with Glove-trained word vectors containing global and local contextual information of a general-purpose corpus, which simultaneously improves the model's translation performance and generalization ability in specialized domains.

First, the word vectors obtained from pre-training on a large-scale monolingual corpus in the specialized domain using Word2vec are used to initialize the embedding matrix of the model. The set of public word vectors pre-trained by Glove on a large-scale generalized monolingual corpus is compared with the word list of the dataset generated

by the translation model, and the corresponding vectors in the embedding matrix are replaced by the Glove word vectors for the common words that co-occur in both. The word vectors for the other words in the dataset word list are kept unchanged and the Word2vec initialized word vectors are still used. This combines the two approaches, initializing common words with Glove and uncommon terminology words with Word2vec, resulting in a good representation of all words in the dataset word list and compensating for the small corpus size and domain mismatch.

III. C. 3.3 Find and Replace Unregistered Words

In order to find and replace the unregistered words (UNK) in the electrical domain corpus due to the specialized terminology, this paper adds the specialized terminology dictionary as a customized segmentation dictionary during the Chinese and English segmentation, which better segments the specialized terminology into a whole vocabulary. At the same time, the term dictionary is used as the external dictionary of the translation system to find and replace the unregistered words in the sentences translated by the model. When carrying out Chinese segmentation, the precise mode of jieba segmentation is adopted, and its customized dictionary function is used to add the term dictionary to the segmentation dictionary, which ensures to a certain extent the professionalism and completeness of the terminology of the professional field as well as the combinability of some special words, and improves its segmentation effect on the professional corpus. In order to correspond to the participle of professional terms in Chinese sentences, this paper uses the MWETokenizer phrase splitter in the NLTK participle tool to set the phrases in the term dictionary as customized words to split English sentences. This allows the model to accurately learn the mapping relationships of phrases in the term dictionary. When translating the i th sentence, each word in the sentence is judged to be in the terminology dictionary as a way of finding out if the terminology exists in the sentence. If it exists, the terminology word in the sentence is added to a temporary list of specialized terms. When an unregistered word appears during the decoding process, it is replaced using the corresponding term word in the list. Such an approach, to some extent, side-steps the problem of recognizing terminology words in sentences. Compared with the method of training and using a named entity word recognition model to recognize professional terms, the method proposed in this paper is more concise and effective, saves a lot of trouble, and ultimately makes the translation model more effective in the professional domain.

IV. Analysis of the effect of English translation modeling

IV. A. 4.1 Experimental data

In order to realize the task of translating professional terms, this paper takes the translation in the field of electrical engineering as an example, and the Chinese-English parallel corpus used comes from professional works on electrical engineering, relevant literature and patent texts in the field of electrical in the past 20 years, and manuals of electrical equipment. There are about 150,000 parallel sentence pairs in the actual training set, 3,000 parallel sentence pairs in the validation set and 3,000 parallel sentence pairs in the test set, and the terminology dictionary used in the experiment contains 7,149 English terminology terms, which are all taken from the training set.

IV. B. 4.2 Assessment of indicators

The experimental results use BLEU to evaluate the quality of translation, which is the most commonly used evaluation index for neural machine translation at present, and it checks the similarity of the multivariate phrases in the actual translation and the reference translation to judge the translation effect, specifically, it counts the minimum value of the number of times the same multivariate phrases appear in the actual translation and the reference translation of each sentence, and then divides it by the total number of multivariate phrases in the translation to get the corresponding precision, and then the corresponding operation is performed on the precision to get the final BLEU value. After that, the final BLEU value is obtained by performing the corresponding operation on the precision. In addition, the BLEU difference between models, the average training time of the models in multiple experiments, and the increase of the training time of the models are also counted in this paper.

IV. C. 4.3 Experimental results

IV. C. 1) 4.3.1 Model training

Taking the pure Transformer model provided by Open-NMT system as the baseline model, the improvement process of the model translation quality is shown in Figure 1. During the training process of 20000 steps, the improvement rate of the translation quality of the two is basically the same, but when the folding line tends to be stable (after 10000 steps), the BLEU value of the English Translation Neural Network model incorporating terminology information in this paper is always better than Baseline, with a BLEU value of about 35%, which indicates that the optimization method proposed in this paper improves the performance of the model consistently and stably.

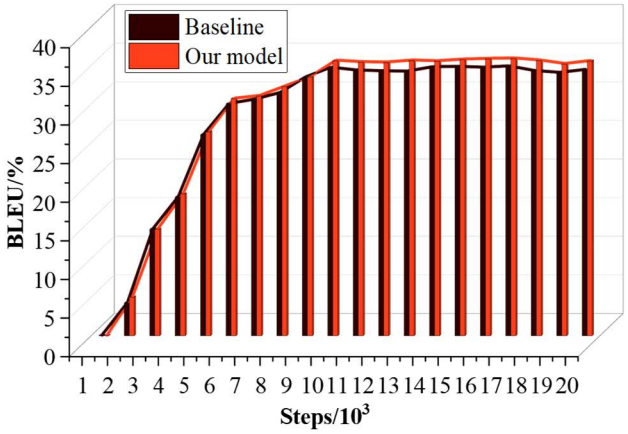


Figure 1: Model translation quality improvement process

IV. C. 2) 4.3.2 Comparison experiments

In order to further validate the effectiveness of the method proposed in this paper, comparative experiments are conducted with other models on an electrical engineering corpus.

Method 1: The method introduces factual relation information as a priori knowledge into the Transformer and proposes a factual relation enhancement method to guide the training of the decoder and eventually fusion in the contextual attention of the decoder.

Method 2: The method applies the ciphertext-based data enhancement technique, which utilizes the encoder to extract feature information on the plaintext and ciphertext of the source language respectively, and fuses them through the attention mechanism, encouraging the model to refer to the more generalized features in the ciphertext to learn richer knowledge in order to improve the quality of the model's translation.

Method 3: This method mixes the a priori translations proportionally into the Transformer, and dynamically adjusts the mixing ratio using learnable parameters to enable the model encoder to learn cross-linguistic knowledge. Its a priori knowledge is processed using a parameter-sharing module, and the step of incorporating a priori knowledge is performed in each sublayer of the encoder.

Method 4: This method builds an additional structure outside the Transformer to get the possible paraphrases of the source language words using a bilingual dictionary, and then incorporates the dictionary information into the translation model using pointers, disambiguators, and replicators to improve the quality of the model's translations.

Method 5: This method proposes a way to inject custom terms into the translation model at runtime by adding annotations to the raw data and splicing the annotation vectors with the word vectors to improve the quality of the translations without changing the structure of the model.

Method 6: Transformer-Big model, which is wider and doubles the number of attention heads, word feature dimensions and the dimensions of feed-forward neural network layers, is a classic Transformer improvement model.

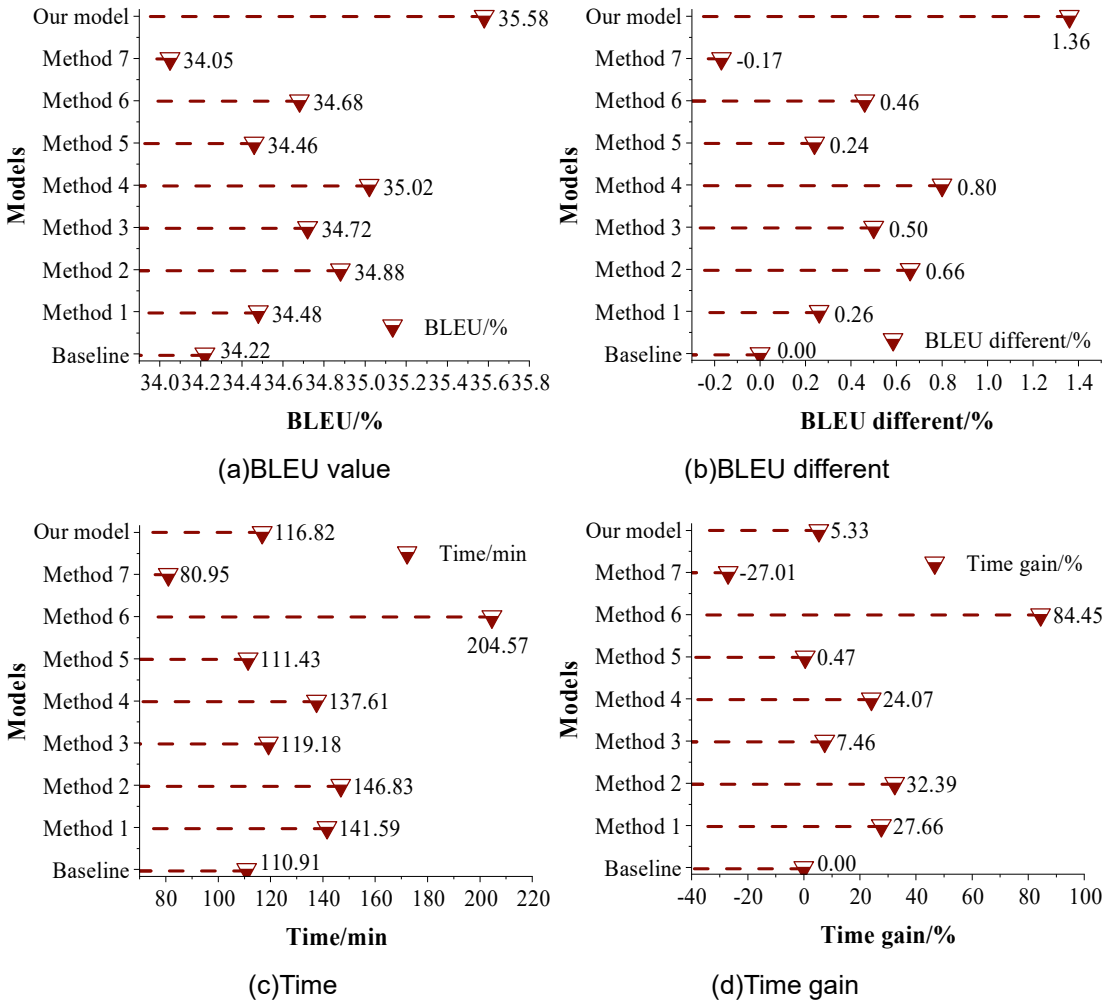
Method 7: This model changes the basic architecture of Transformer by using a dual attention module in the encoder sublayer, while reducing the number of feedforward neural network layers and the number of decoder sublayers.

The results of the comparison experiments of different translation models are shown in Fig. 2, and (a)~(d) are the BLEU value, BLEU difference, training time and time rise of each model, respectively.

It can be found that Method 4 does not utilize the original structure of Transformer, but repeatedly applies the attention module and feed-forward neural network in the additional pointer, disambiguator, and replicator for information extraction and fusion, which adds a large number of learnable parameters to the model. Method 1 and Method 2 use encoder parameter sharing to minimize the number of parameters in the model, but the fusion of prior knowledge is still achieved by adding a new contextual attention module in the decoder. These additional parameters improve the expressive power of the model while potentially increasing the risk of overfitting, making it difficult to adequately train the model for each parameter under low-resource conditions, and the additional parameters cause the training time to rise. As a result, all three neural network translation models have a high rise in time spent, ranging from 24.7% to 32.39%. Method 5 achieves the improvement of translation quality by only adding annotations to the terms, without introducing new learnable parameters or changing the structure of the Transformer, so the training time is basically the same as Baseline, and the rise in the time spent is only 0.47%, but this method has no terminology information extraction process, and it requires the model to autonomously learn the

meanings and features of the annotations, which is limited to the performance improvement of the model under the low-resource conditions, this method has limited improvement in model performance. It can be seen that the quality of English translation within the electrical engineering domain of these comparative models introducing prior knowledge is not as good as that of the neural network model proposed in this paper, with BLEU values less than 35.1%.

From the experiments of Method 6 and Method 7, it can be seen that although increasing the width of the model can improve the model performance and the quality of translations to a certain extent, it also increases the training time significantly, Method 6 takes the longest time, 93.66 minutes more than Baseline, with an 84.45% increase in time, but the BLEU is not as good as that of this paper's English translation neural network model, the difference in BLEU between the two is 0.46% and 1.36% respectively. It can also be noticed that these more classical improvement methods in the general purpose domain have limited performance improvement for the task of translating specialized terminology, and the BLEU of Method 7 not only does not align with Baseline, but even decreases by 0.17%.



(a)BLEU value (b)BLEU different

(c)Time (d)Time gain

Figure 2: Comparison of different translation models

IV. C. 3) 4.3.3 Source Language Length Effects

In order to further verify the performance of the model in the task of translating sentences with different lengths, the study analyzes the lengths and BLEU metrics of the source language as well as the translated language with different lengths in the dataset and compares them with the Trasformer model, and the results of the lengths of the translations with different source language lengths are shown in Figure 3. With the increase of the source language length, the length of the translated language also increases gradually, and its length is more consistent with the source language length interval. The results of the metrics comparison under different source language lengths are shown in Figure 4. With the increase of source language length, the BLEU index of the research model gradually

decreases, but the BLEU index of this paper's model under different source language lengths is greater than that of Trasformer model, and the overall BLEU index reaches 27.75%, while the overall BLEU index of the Trasformer model is 23.17%, compared to which, the BLEU value of this paper's English-Translation neural network model incorporating terminology information is improved. In comparison, the BLEU value of the English translation neural network model incorporating terminology information in this paper is improved by 4.58%, which indicates that the proposed model has better quality of terminology translation.

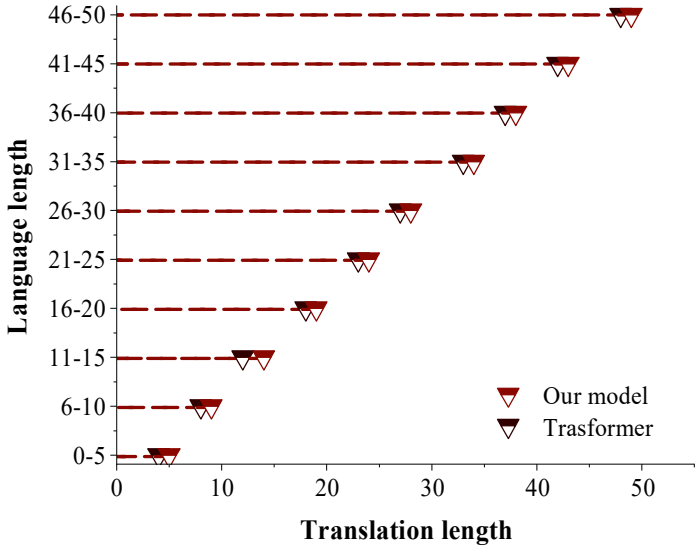


Figure 3: Translation length of different source language length

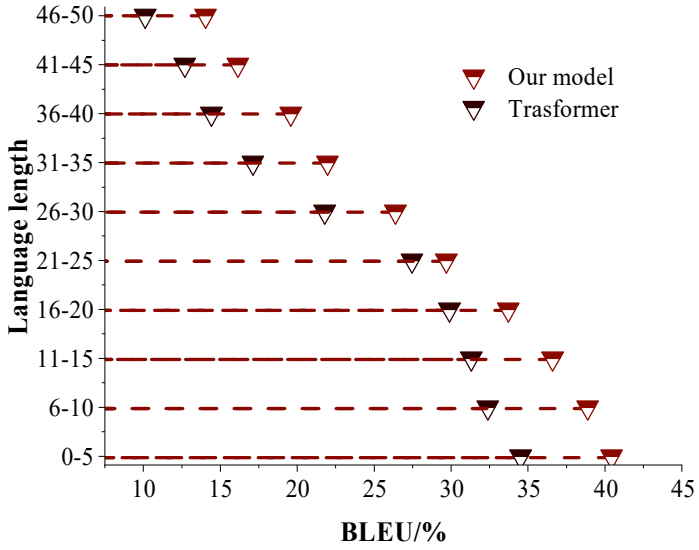


Figure 4: The comparison results of different source language length

V. Conclusion

The English translation neural network model incorporating terminology information effectively improves the translation quality of professional terminology through a multi-level optimization strategy. Experiments on about 150,000 parallel corpora in the field of electrical engineering prove that the method performs well in source language tests of different lengths, with an overall BLEU value of 27.75%, which is 4.58 percentage points higher than the 23.17% of the Transformer model. What's more, the present model outperforms the baseline model in all source language length intervals, showing a stable performance improvement. Compared with Method 4, Method 1 and Method 2, which introduce complex structures, the training time of these methods rises by as much as 24.7% to 32.39%, while the present model only increases the training time by 5.86%. Comparing with Method 6 (Transformer-Big model), despite its increase in model width, number of attention heads and feed-forward neural

network layer dimensions, the training time hike is 84.45%, while the BLEU enhancement is inferior to that of this paper's method. The experimental results fully demonstrate that the combined application of term dictionary-assisted word segmentation, word vector pre-training by combining Glove and Word2vec, and the unregistered word finding and replacing mechanism can effectively improve the quality of text translation in specialized domains without significantly increasing the computational cost. This research provides new ideas for solving the term translation challenges in vertical domain machine translation, and also provides a referenceable method for professional term translation in other domains.

Funding

This research was supported by the 2024 Key Project of Humanities and Social Sciences Research in Anhui Province's Universities: “Research on the External Promotion and Translation of Cultural Heritage in Maanshan Area from the Perspective of Ecological Translation Studies” (2024AH053311).

References

[1] Polyakova, L. S., Yuzakova, Y. V., Suvorova, E. V., & Zharova, K. E. (2019, March). Peculiarities of translation of English technical terms. In IOP Conference Series: Materials Science and Engineering (Vol. 483, No. 1, p. 012085). IOP Publishing.

[2] Khrystiuk, S. (2019). Specifics of english economic terminology translation. International journal of philology, 1(23), 94-99.

[3] Palumbo, G., & Duin, A. H. (2024). The evolutionary convergence of technical communication and translation: An integrative literature review of scholarship from 2000 to 2022. IEEE Transactions on Professional Communication.

[4] Zheng, W. (2017, February). Translation strategies for texts of science and technology. In 2017 International Conference on Humanities Science, Management and Education Technology (HSMET 2017) (pp. 32-36). Atlantis Press.

[5] Mulyanah, A. (2020, April). The strategy of terminology translation. In Twelfth Conference on Applied Linguistics (CONAPLIN 2019) (pp. 1-5). Atlantis Press.

[6] Olalla-Soler, C. (2019). Using translation strategies to solve cultural translation problems. Differences between students and professional translators. Perspectives, 27(3), 367-388.

[7] Zeffiro, V., Sanson, G., Vanalli, M., Cocchieri, A., Ausili, D., Alvaro, R., & D'Agostino, F. (2021). Translation and cross-cultural adaptation of the Clinical Care Classification system. International Journal of Medical Informatics, 153, 104534.

[8] Kuzenko, H. (2017). Translation as cross-cultural communication. Science and education a new dimension. Philology, 33(123), 40-43.

[9] DI, H. (2022). Role of Translation as a Professional Service in Cross-Cultural Communication: A Review of Translator Status and Professional Career Planning. Higher Education and Oriental Studies, 2(3).

[10] Al-Musawi, N. M. (2014). Strategic use of translation in learning English as a Foreign Language (EFL) among Bahrain university students. Comprehensive Psychology, 3, 10-03.

[11] Qassem, M. (2021). Translation strategy and procedure analysis: a cultural perspective. Asia Pacific Translation and Intercultural Studies, 8(3), 300-319.

[12] Bi, S. (2020). Intelligent system for English translation using automated knowledge base. Journal of Intelligent & Fuzzy Systems, 39(4), 5057-5066.

[13] Haque, R., Hasanuzzaman, M., & Way, A. (2020). Analysing terminology translation errors in statistical and neural machine translation. Machine Translation, 34, 149-195.