electronics

Article

Part-of-Speech Tags Guide Low-Resource Machine Translation

Zaokere Kadeer †, Nian Yi *,† and Aishan Wumaier

Xinjiang Laboratory of Multi-Language Information Technology, School of Cyber Science and Engineering, Xinjiang University, Urumqi 830046, China; zuhra@xju.edu.cn (Z.K.); hasan1479@xju.edu.cn (A.W.)
* Correspondence: ynian@xju.edu.cn
† These authors contributed equally to this work.

Abstract: Neural machine translation models are guided by loss function to select source sentence features and generate results close to human annotation. When the data resources are abundant, neural machine translation models can focus on the features used to produce high-quality translations. These features include POS or other grammatical features. However, models cannot focus precisely on these features when data resources are limited. The reason is that the lack of samples makes the model overfit before considering these features. Previous works have enriched the features by integrating source POS or multitask methods. However, these methods only utilize the source POS or produce translations by introducing the generated target POS. We propose introducing POS information based on multitask methods and reconstructors. We obtain the POS tags by the additional encoder and decoder and compute the corresponding loss function. These loss functions are used with the loss function of machine translation to optimize the parameters of the entire model, which makes the model pay attention to POS features. The POS features focused on by models will guide the translation process and alleviate the problem that models cannot focus on the POS features in the case of low resources. Experiments on multiple translation tasks show that the method improves 0.4∼1 BLEU compared with the baseline model on different translation tasks.

Keywords: neural machine translation; part of speech; loss function; low resource translation; reconstructor; multitask

Citation: Kadeer, Z.; Yi, N.; Wumaier, A. Part-of-Speech Tags Guide Low-Resource Machine Translation. *Electronics* **2023**, *12*, 3401. https://doi.org/10.3390/electronics12163401

Academic Editor: Arkaitz Zubiaga

Received: 18 May 2023
Revised: 20 July 2023
Accepted: 4 August 2023
Published: 10 August 2023

1. Introduction

With the rapid development of neural networks, neural machine translation models [1–5] have achieved impressive results, which mainly employ an end-to-end framework consisting of an encoder–decoder for translation. The encoder extracts the semantic features of the source language sentences. The tokens of the target language are predicted by the decoders based on the features proposed by the encoder. Therefore, the ability of the encoder to extract linguistic information and the ability of the decoder to infer the tokens of the current moment determines the translation model performance. The model has relatively enough data for translation tasks with rich data resources to generate better results. Therefore, language information extracted by the encoder and decoder is often more comprehensive in training. However, for translation tasks with scarce data resources, the model needs to generate results with limited data, which may cause the model to reduce its attention to a part of the information, resulting in imperfect features extracted by the model and poor robustness.

To enable the model to capture more information, some works attempt to integrate syntactic information into the translation model. These works can be divided into three categories. Refs. [6–16] These works enrich the features extracted by the encoder to improve the performance of the translation model. These works show that fusing grammatical information inside machine translation can effectively improve the performance of translation models. Also, these approaches show that the encoder features must contain POS (part of speech) information as long as the POS information is spliced to the source. However,

based on other grammatical annotation tasks, such as POS [17–19], we believe sentences already contain grammatical information, because the input to the lexical annotation task is sentences. In addition, the features extracted by the pretrained model [20,21] are used for various tasks, which also suggests that sentences contain various information. However, we are not determining whether this direct addition of grammatical information to the model will lead to an excessive focus on grammatical information. After all, the model cannot extract the required features precisely because of the amount of data. Also, most of these methods can only add the source grammatical information. And the grammatical information of the target-side sentence may differ from that of the source-side sentence. However, the inability to introduce the target's grammatical information and the decoder's inability to extract the grammatical information accurately may lead to the negative impact of this difference on the model. Refs. [22–24] used multitask learning to introduce language knowledge into RNN-based machine translation models and improved the performance of NMT (neural machine translation) by exploiting the prediction results. Multitasks are also used for machine translation and other tasks [25–27]. Refs. [24,28] improve machine translation performance by exploiting the predicted target-side language information. These multitasking methods also prove that sentences contain a variety of syntactic information. Whether fully shared or semishared parameter multitasking methods [22], their inputs are sentences. But their output can be a target-side sentence or a POS. We consider the process of model training as the process of feature extraction and feature selection. The fine-tuning of the pretrained model can be seen as selecting features for the task. Therefore, the multitask approach is preferable to the previous one. And the composition of the loss function is crucial to the multitask approach. We consider that the difference in features is caused by the target sequences of different tasks (the target sequence of the POS tagging task [17–19] is the POS tag, while the target sequence of the machine translation task is the target end sentence). Therefore, the difference in the loss of the models leads to a difference in the features that the models focus on because the information in the sentences is too complex. The model can only extract the valuable features as comprehensively as possible. However, when data resources are abundant, the model can extract features that contain grammatical knowledge through the data, such as part of speech. Multitasking approaches using parallel or shared decoders can alleviate these problems, but most current multitasking approaches use syntactic features from only one end. Predictive and syntactic features of sentences are used to help translation models. Although this approach can use source- and target-side syntactic features, it increases the model inference time. The grammatical features obtained by layer-by-layer prediction accumulate various errors during propagation. Besides, data augmentation [29–31] and pretrain [20,21] model are used to improve the performance of NMT.

In summary, we argue that NMT can extract grammatical features from a large number of sentences. The loss of NMT is used to guide NMT to extract various features in the sentence so that the model can use these features to obtain better results. We add a loss function to optimize the model based on a multitask approach. However, methods that share encoder and decoder parameters can interfere with the target translation results when using source lexical information. This interference is not mitigated by weighting the losses differently. Therefore, we added a reconstructor to the model. The reconstructor approach [32] ensures that the translation result is consistent with the semantics of the input text. Therefore, the introduced reconstructor can ensure that POS information is included in the output of the features by the encoder and decoder. Further, it also can determine that the POS information in these features is consistent with the POS sequence information. Therefore, we make NMT pay attention to the POS features contained in sentences by introducing a loss function of the POS. These POS features include the POS of the current token and the parts of speech of adjacent characters. Because POS sequences are the same as sentence sequences, the tags of POS should not be independent. The parts-of-speech loss function is used to make models pay attention to the POS information and gain the relationship between the tags of the POS. This method allows us to utilize the

POS features at the source and target. Since the additional structure in the training process is not used in prediction, the time for inferring sentences remains the same. Meanwhile, the loss function in our approach consists of several components, so we can weigh the different loss functions to ensure that the impact of the loss models differs for different tasks. The overall diagram of the method is shown in Figure 1. Our contributions are as follows:

(1) We set different weights for different loss functions, which allows us to distinguish the task's levels according to the task requirements.
(2) We use a reconstructor approach to ensure consistency between text features and lexical information.
(3) Our approach does not increase inference time when introducing multiterminated lexical information.

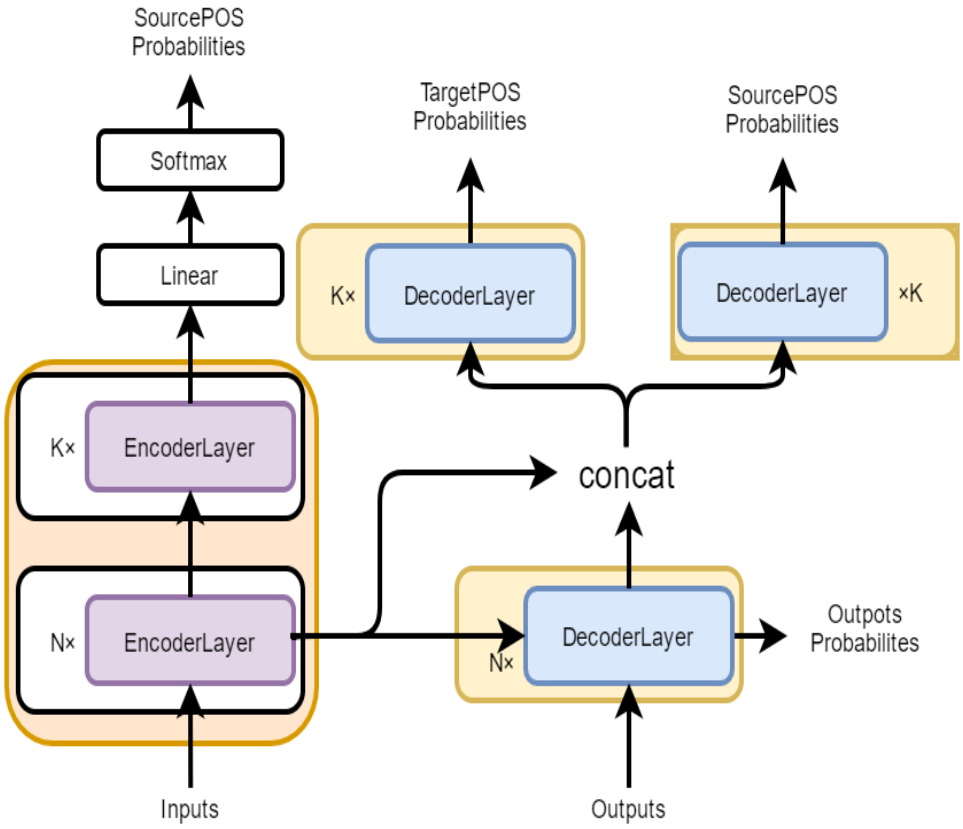

Figure 1. The overall diagram of the method.

The remaining sections of this paper are organized as follows: In Section 2, related work on low-resource machine translation and the introduction of grammatical information into machine translation is reviewed. Our model is based on the transformer, so we briefly describe its mechanism in Section 3. Section 4 describes our approach. The data and model parameters used for the experiments, the experimental results, and the analysis are shown in Section 5. We conclude with a summary in Section 6.

2. Related Work

To make NMT capture more information, a large body of work attempts to incorporate syntactic information into NMT to enrich the information extracted by NMT [6,7]. The method of integrating linguistic information into the source side enriches the features extracted by the encoder to improve the performance of the translation model. Sennrich and Haddow [8] used grammatical knowledge such as word roots, subword tags, morphological feature information, POS tags, and dependency analysis tags to improve the performance of translation models. They used word embedding layers to integrate dif-

ferent grammatical knowledge and character features to enrich the encoding-extracted features. Eriguchi et al. [6] proposed a tree-based encoder, which processes sentences from the bottom up in the form of a tree to introduce the information of the dependency tree and enriches the extracted features after combining with the original encoder to improve the model performance. Chen et al. [9] proposed a bidirectional dependency tree encoder based on Eriguchi, which processes source language sentences through bottom-up and top-down approaches. Unlike the previous methods, Bastings et al. [10] extracted the features of source language sentences by introducing syntactic graph convolution and used this method to introduce the information of a syntactic dependency tree. Hashimoto and Tsuruoka [11] let the model learn the graph structure hidden in the sequence by changing the form of the input sequence. Li [12] integrates syntactic information by combining tokens with syntactic tags through parallel, hierarchical, and hybrid methods. In [13], the performance of translation models is improved by adding encoders and decoders to encode source-side and model-target-side dependency information. Zhang et al. [14] used the output of the encoder to predict the target-side syntactic information. They then concatenated it with the features of the target-side tokens to integrate the syntactic information. Bugliarello and Okazaki et al. [15] proposed dependency-aware self-attention, which uses Gaussian distribution to weigh the dependency information to weight the attention matrix of tokens. Chakrabarty et al. [16] used language features or word embeddings of tokens to weight language features and integrated self-attention and language features based on token attention into the translation model. Ref. [33] proposes a neural network model that combines source-side syntactic knowledge and multiheaded self-attention to make the syntactic part more attentive. They produce a bias by an additional syntactic perceptron and use that bias to correct the attention distribution. Meaningful semantic features in the source sentences are also obtained through a random discard strategy. Ref. [34] applies a combination of multiple dictionary-based data enhancement techniques to low-resource machine translation. They introduce monolingual data to bilingual word embedding learning to reduce the out-of-vocabulary problem. To overcome the limitations of sentence representation reasoning, proposes a deep fusion matching network that consists of a coding layer, matching layer, dependency convolution layer, information aggregation layer, and inference prediction layer. The matching layer is enhanced using a deep matching network, and a heuristic matching algorithm simplifies the interactive fusion. The dependency convolution layer utilizes a tree-type convolution network to extract sentence structure information, improving the interpretability of the reasoning process. Experimental validation is conducted on multiple datasets to assess the performance of the proposed model. A hybridized form of direct and rule-based language processing is used in [35] to present a machine translation system from Sanskrit to Hindi. The divergence between Sanskrit and Hindi is also discussed in this paper, along with a proposition for how to handle it. Sanskrit–Hindi bilingual dictionaries, a grammatical Sanskrit corpus and a Sanskrit analysis rule base have all been used in the projected system. The system processes the input Sanskrit sentence using the parsing approach and the context-free grammar in standard form for Sanskrit language processing. Ref. [36] focuses on integrating linguistic features into a neural machine translation system, especially for low-resource languages such as Thai and Burmese. The proposed approach is carried out by feeding grammatical features into the decoder and text features after extracting the features through the encoder.

Niehues and Cho [22] used multitask learning to introduce language knowledge into RNN-based NMT by jointly training several natural language processing (NLP) tasks in one model to improve the performance of a single task. On this basis, Yin et al. [24] enriched the feature vectors extracted by the encoder by introducing the predicted POS tags of the source language sentence sequences and improved the translation model's performance by incorporating the predicted POS tags of the target-side sentences. In [24,28], machine translation performance is improved by predicting target language information. Maimaiti's [37] method is equal to the previous ones, and they combine POS tags and data augmentation methods with improving translation performance by obtaining a large amount of generated data.

Ref. [38] designs a deep fusion matching network to solve the problem of insufficient inference depth and interpretability in reasoning models. Ref. [39] proposes a multilayer semantic-based sentence representation method using a bidirectional long- and short-term memory network and a multiattention mechanism. Unlike the introduction of grammatical information in text, ref. [40] enriches the information at the source end by incorporating the extracted image features. They combine image and text features and then make a self-attentive mechanism with text features. Ref. [41] extends the parent–child transfer learning method using embedded repetitions between aligned subwords to improve low-resource machine translation. Ref. [42] quantifies the data produced by back translation through lexical diversity and syntactic diversity and improves the effectiveness of back translation by enhancing the diversity of produced translations. Ref. [43] compares the effects of syllables, graphemes, and characters as inputs on model performance. The model uses regularization to syllabify some languages and hyphenation as a substitute for languages that cannot be syllabified. Ref. [44] proposes a series of values that are particularly suitable for low-resource environments, emphasizing that default or recommended values for high-resource environments are suboptimal for low-resource environments. They argue that a more aggressive regularization is necessary when resources are limited in proportion to their scarcity. The authors explain their findings by considering the generalization ability of sharp and flat basins in neural network loss landscapes.

3. Background

Since the model is built upon the Transformer [3], we introduce the architecture of the Transformer in this section. We denote the source sentence as $X = \{x_1, x_2, x_3, ..., x_n\}$ and the target sentence as $Y = \{y_1, y_2, y_3, ..., y_m\}$, where n and m are the lengths of X and Y, respectively. The Transformer network structure is shown in Figure 2.

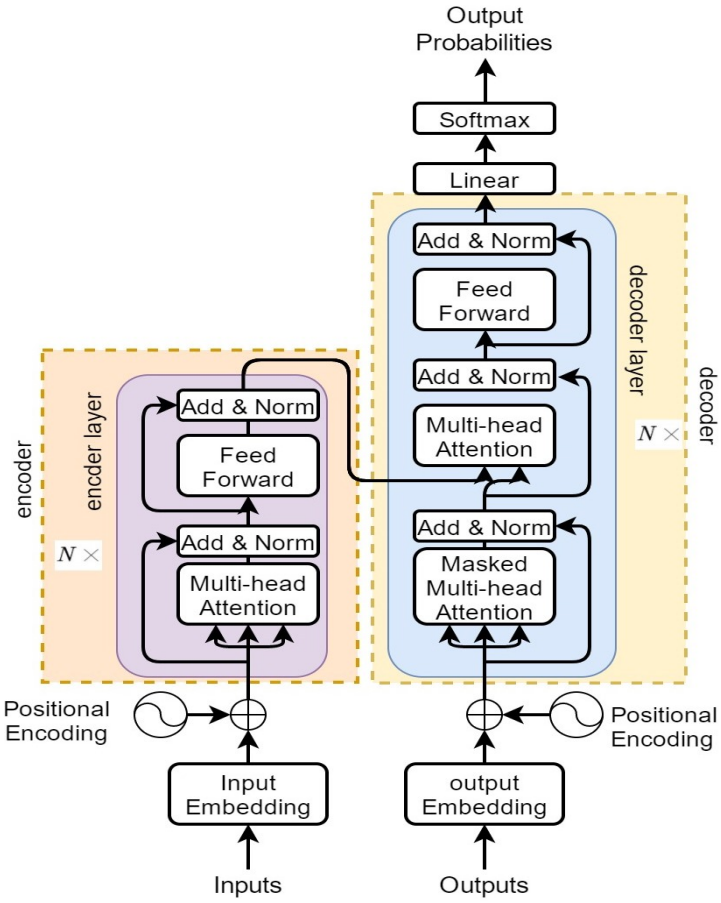

Figure 2. The model structure of the Transformer [3].

3.1. Encoder

The N encoder layers make up the encoder. The text sequence needs to pass through the word embedding layer before it enters the encoder. Each encoder layer is composed of two network structure: (1) the multihead attention network ($MultiHead$) and (2) the feed-forward network (FFN). The information in the sequence is extracted from the sequence by the multiheaded attention network through self-attention. Features are combined and nonlinearly mapped as they pass through the feed-forward network. The output of layer k is then fed to the layer $k+1$ as the input as follows:

$$s_k = EncoderLayer(s_{k-1}) \tag{1}$$

where s_k is the output of k-th layer. Consequently, the encoder layer consists of multihead attention and a feed-forward neural network:

$$EncoderLayer(s_{k-1}) = FFN(MultiHead(s_{k-1})) \tag{2}$$

Multihead and FFN are as follows:

$$FFN(s_k) = \max(0, s_k W_1 + b_1)W_2 + b_2 \tag{3}$$

$$MultiHead(s_{k-1}) = Concat(head_1, head_2, ...head_h)W^m \tag{4}$$

where each head is the output of a self-attention layer:

$$head_i = Attention\left(s_{k-1}W^k, s_{k-1}W^q, s_{k-1}W^v\right) \tag{5}$$

$$Attention(k, q, v) = Soft\max\left(\frac{qk^T}{\sqrt{d_k}}\right)v \tag{6}$$

We denote the output of N-th layer as s, which is the extracted feature of the sentences. The whole process is simplified as follows:

$$s = Encoder(X) \tag{7}$$

3.2. Decoder

The decoder architecture is similar to the encoder, which is also composed of N layers with the same structure. The decoder layer has an additional multiheaded cross-attention sublayer compared with the encoder layer. The process of the decoder is:

$$c_k = DecoderLayer(c_{k-1}, s) \tag{8}$$

where c_k is the output of the k-th layer and s is a feature of a source sentence extracted by the encoder. Finally, the softmax layer is used to output the probability distribution of target words, and the model is trained by the cross-entropy loss, which minimizes the negative log-likelihood as

$$\mathcal{L}_{NMT} = -\sum_{i=1}^{m} \log p(y_i | y_{<i}, X) \tag{9}$$

where y_i is the i-th word in the target sentence Y, and $y_{(<i)}$ is the previous words in the target sentence. In this paper, we denote the decoding process of the NMT model to

$$Y' = Decoder(s) \tag{10}$$

where Y' is translation result of NMT.

4. Methodology

Our model uses POS tags to guide translation. Specifically, the final output of the translation model is guided by the source and target POS losses. Since the model can utilize both the source-side and target-side POS tags, this paper introduces the method from source-side- and target-side-guided translation.

For the guidance of source POS tags, we use an encoder and decoder to calculate the loss of source POS tags; this loss make the model notice POS features during training. The encoder and decoder structure of this method is shown in Figure 3. The sublayer structure of *pos_encoder* and *pos_decoder* is the same as the structure of the encoder and decoder in the Transformer. The difference is the number of sublayers. Considering that the POS tagging task is simpler than the machine translation task, the number of layers of *pos_encoder* and *pos_decoder* is lower than the number of layers in the encoder and decoder.

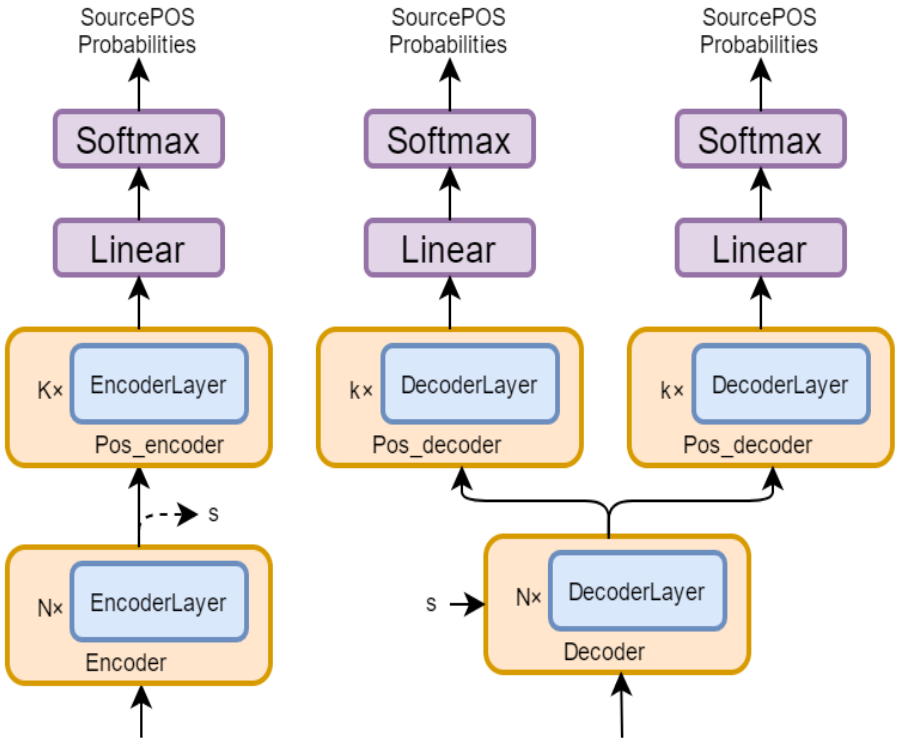

Figure 3. The structure of the encoder and decoder in our method.

The encoder in this structure is used to obtain the context vectors of the source sentence. This process is the same as the encoder process in the original Transformer, and then s will be input into the decoder and *pos_encoder* for translation tasks and POS tagging tasks. The target-side sentence features obtained by decoder inference are input into *pos_decoder* to predict the source-side POS tags. For *pos_encoder*, the context vector s output from the *encoder* is processed by multihead attention to obtaining the features of the source POS sequence:

$$c^{spos} = Pos\_encoder(s) \tag{11}$$

s is the context feature vector extracted by the encoder; $c^s pos$ is the feature vector of the source POS tag obtained by *pos_encoder* through s; and the processing flow of $Pos\_encoder()$ is

$$Pos\_encoder(s) = FFN(MultiHead(s)) \tag{12}$$

The feature is processed by a linear function and a softmax function to obtain the probability of the source language POS tag:

$$p(x^{pos}) \propto \exp(Linear(c^{spos})) \tag{13}$$

Pos_decoder will predict the remaining part of the source POS tags according to the partial tags of the source-end POS and the target-end context information c inferred by the decoder:

$$c^{spos} = Pos\_decoder(x^{pos}, c) \tag{14}$$

where x^{pos} is part of the source POS tag, c is the context feature of the target-side sentence obtained by decoder inference, and c^{spos} is the probability of the source-end POS tag predicted by *pos_decoder*. The use of the target-side POS tag is the same as in the source-side method. There is no *pos_encoder* to predict the target-side POS feature, and only a *pos_decoder* is used to predict the target-side POS tag.

According to the model structure and the description, the objective function of the model proposed in this paper consists of five parts: the loss $\mathcal{L}^{spos}$ obtained by the probability of the POS tag output by the encoder, which is obtained through the sentence sequence contained in the semantically computed.

$$\mathcal{L}^{spos} = -\sum_{i}^{n} \log p\left(x_i^{pos} | x\right) \tag{15}$$

Losses $\mathcal{L}^{spos'}$ and $\mathcal{L}^{tpos}$ are computed from the output of *pos_decoder*. This partial loss is computed jointly from the source and target sentence sequences and the known POS sequences. To enable the model to learn the POS conversion between language pairs, $\mathcal{L}^{spos'}$ is added on the decoder side.

$$\mathcal{L}^{spos'} = -\sum_{i}^{n} \log p\left(x_i^{pos} | x, y, x_{<i}^{pos}\right) \tag{16}$$

$$\mathcal{L}^{tpos} = -\sum_{i}^{m} \log p\left(y_i^{pos} | x, y, y_{<i}^{pos}\right) \tag{17}$$

The loss $\mathcal{L}^{word}$ calculated by the output of the *decoder*:

$$\mathcal{L}^{word} = -\sum_{i}^{m} \log p(y_i | x, y_{<i}) \tag{18}$$

The KL divergence $\mathcal{L}^{KL}$ is calculated by the source POS sequence probability obtained by the *pos_encoder* and the source-end POS sequence probability output by *pos_decoder*. Since$\mathcal{L}^{spos}$ and $\mathcal{L}^{spos'}$ have different information, the difference between the POS features extracted by the decoder and the encoder is determined by KL divergence.

$$\mathcal{L}^{KL} = \sum_{x_i^{pos} \neq x'^{pos}_i} D_{kl}\left(\log p\left(x_i^{pos} | x\right) || \log p\left(x'^{pos}_i | x_{<i}^{pos}, x, y\right)\right) \tag{19}$$

5. Experiments

5.1. Dataset

The train set use in the Chinese–Uighur and Uighur–Chinese experiments have 170,000 sentences provided by the CCMT2019 (China Conference on Machine Translation) Uighur–Chinese machine translation task. The test sets and validation sets have 1000 sentences provided by CCMT2019. We use word segmentation tools and POS tagging tools developed by Xinjiang Multilingual Information Technology Laboratory to process the Uighur language. For Chinese, we use the open-source tool THULAC [45] of Tsinghua University for word segmentation and POS tagging. We also use BPE [46] (Byte-Pair Encoding) to process Chinese and Uighur with 8K iterations.

The data use in the Chinese–English and English–Chinese experiments have 230,000 sentences provided by IWSLT2017 (https://workshop2017.iwslt.org/index.php (accessed on 9 March 2022)) (The International Conference on Spoken Language Trans-

lation). The validation set is IWSLT17.TED.dev2010 and the test set is IWSLT2017.TED-.tst2015. We used THULAC (Tsinghua University Lexical Analyzer for Chinese) and NLTK (https://www.nltk.org/ (accessed on 9 March 2022)) (Natural Language Toolkit) to perform word segmentation and POS tagging on Chinese and English data, respectively. The English–Russian and Russian–English experiments are performed on 1.55 million sentences provided by WMT14 (https://www.statmt.org/wmt14/translation-task.html (accessed on 9 April 2021)) (Workshop on Machine Translation), where the validation set is newstest2013, and the test set is newstest2014. The word segmentation and POS tagging for English and Russian uses NLTK. The data use in the German–English direction experiment is the English–German data provided by WMT16 (https://www.statmt.org/wmt16/translation-task.html (accessed on 9 April 2021)). The training dataset has 4.52 million sentences, and the validation and test sets are newstest2015 and newstest2016. Spacy (https://spacy.io/ (accessed on 15 May 2022)) is used to perform word segmentation and POS tagging for English and German. The above experimental data are all processed by BPE with 32K iteration. The translation results are calculated using Moses's (https://www.statmt.org/moses/ (accessed on 1 March 2019)) evaluation script multi-bleu.perl to obtain the BLEU [47] value. Among them, the F1 value of the POS tagging tool (THULAC) can reach 92.9% on the standard dataset Chinese Treebank (CTB5) (https://www.cs.brandeis.edu/~clp/ctb/ (accessed on 15 March 2022)); the accuracy rate of spacy in the current language reaches more than 95%. The specific information of the data is shown in Table 1. We used Chinese–Uighur and Chinese–English as low-resource translation tasks and the other two as resourceful translation tasks. In this paper, Chinese is represented as zh, and English, Uighur, and Russian are denoted as en, uy, and ru, respectively.

Table 1. Experimental dataset, zh means Chinese.

Language Pair	Source	Train Data Size	Valid Data Size	Test Data Size
zh-uy	CCMT2019	170,054	1000	1000
zh-en	IWSLT2017	231,230	879	1000
en-ru	WMT2014	1,558,847	3000	3000
de-en	WMT2016	4,500,966	2169	2999

5.2. Hyperparameters and Systems

The base system is trained using the open-source system Fairseq (https://github.com/pytorch/fairseq (accessed on 15 March 2020)), and other models are modified on Fareseq. And the settings of the model hyperparameters are as follows:

- The head is 8 in multihead attention.
- Encoder and decoder layers are 6.
- The dimension of the word vector and model hidden state is 512.
- The dimension of the feed-forward network is 2048.
- The optimization function is Adam [48], where adam_beta1 is 0.9 and adam_beta2 is 0.998.
- Dropout [49] is 0.1.
- The warm up [50] is 4000.
- The learning rate is 0.0007.
- The label_smoothing is 0.1.

In the decoding stage, Beam Search [51] is used to search, the number of search candidates is 24, and the length ratio of the translated target sentence to the source sentence is 1.6.

Fairseq is used for training, and the following systems are used for experiments:

- `base`: The base system is based on the Transformer base parameter.
- `Sennrich [8]`: Since the experiments performed by Sennrich [8] are based on the RNN machine translation model, we reproduce the method in the Transformer according to the method proposed in Sennrich's paper, and this system is used as Sennrich. This

approach enriches the feature information of the encoder by splicing POS information with word vector information.

- `Niehues [22]`: The methods proposed in [22] is implemented and experimented with transformers, denoted as Niehues. This method adds POS tagging to the machine translation task but uses only the POS labels at one end.
- `Yin [24]`: The methods proposed in [24] is implemented and experimented with transformers, denoted as Yin. The method is similar to Niehues, except that the decoder is assisted in generating the results by predicting the target point POS.
- `Chakrabarty [16]`: The methods proposed in [16] are implemented and experimented with transformers, denoted as Chakrabarty. The method weights the features when fusing the source-side POS to the encoder.
- `joint`: According to the previous experimental results and Sennrich's method, we define the text's word vector and the word vector of the part of speech as 512 dimensions. After splicing into a 1024-dimensional feature, a linear layer converts the 1024-dimensional feature into a 512-dimensional feature (system joint).
- `srcPOS`: srcPOS is a part of the loss function in our proposed model (the loss function is $L = L^s pos + L^s pos' + L^K L + L^w ord)$); the system only includes the loss of the source-side POS information.
- `tarPOS`: tarPOS is a loss that only contains the target POS information.
- `joint + tarPOS`: joint + tarPOS is the fusion of the modified Sennrich method and the tarPOS method.
- `srcPOS + tarPOS`: srcPOS + tarPOS is a fusion of corresponding system methods.

All results evaluate the translation by calculating BLEU using multi-bleu.perl after averaging 10 models.

5.3. Low-Resource Translation Tasks

Table 2 shows the results of different systems on different translation tasks. Our proposed method has a significant improvement compared with base and other systems. The Sennrich and Niehues methods integrate various grammatical information into NMT, while the methods implemented in this paper only use POS information. We believe that the more syntactic information provided to NMT, the better the performance of NMT can be. Here, we only verify the effectiveness of our method in terms of parts of speech. At the same time, we find that the results of the joint system are significantly better than Sennrich. We believe that the features of the output text of the word embedding layer and the features of the POS tags contain some of the same information. It is difficult for NMT to utilize the most beneficial information of each by just merging the text and POS embeddings. And unlike the original Sennrich, which uses various grammatical information, here it only uses POS tags, and the dimension of the POS tags is small, making NMT unable to obtain enough grammatical information from this simple method.

Table 2. Experimental results on low-resource translation tasks.

System	zh2uy	uy2zh	zh2en	en2zh
base [3]	22.71	27.36	21.25	19.33
Sennrich [8]	22.87	26.92	21.26	19.23
Niehues [22]	23.34	27.86	21.7	19.73
Yin [24]	23.12	27.75	21.62	19.6
Chakrabarty [16]	23.34	26.96	21.67	19.45
joint	23.43	28.14	21.82	19.94
srcPOS	23.36	27.72	21.86	19.79
tarPOS	23.47	28.2	21.86	19.62
joint + tarPOS	23.7	28.39	23.00	20.26
srcPOS + tarPOS	23.88	28.37	22.27	20.46

The best results on the low-source translation tasks are with the method that fuses target and source parts of speech. These results show that the method proposed in this paper to guide NMT to pay attention to the source and target POS information is beneficial to improving the translation model's performance. Our approach is quite different from multitask approaches that parallel or even share decoders; our method indirectly affects the NMT model by iterating the POS tagging task decoder. During inference, the model parameters do not take advantage of the iterative decoder. Therefore, they do not incur additional runtime. We found that the improvement of the model through the joint system is more significant than that of the indirect use of the loss function to implicitly influence the model, which may be related to the depth of the model, which reduces the influence of the loss function. The method proposed in this paper also improves the translation model to a certain extent for the translation task with rich resources.

We also experiment with source-side POS tags in a multitask approach. However, when we take the source POS tagging task, the translation result is far from the ground-truth value, and its BLEU value is even less than 1.0. Therefore, we did not add this method to the results, and for this reason, we use the serial method for POS tagging tasks. We believe that there is a big difference between the source-side POS tags and the target-side sentences of machine translation. This difference makes NMT pay attention to the source-side POS features and reduce the attention to the target-side sentences, resulting in poor translation results. We address the problem of NMT overfocusing on POS information with an additional two-layer encoder sublayer. The feature received by the decoder is still the output of the result of the first part, and the new part will indirectly affect the output of the first part through the objective function. It will not make the model pay too much attention to POS information and reduce the model's performance.

5.4. Rich-Resource Translation Tasks

Compared with the results on low resources, on the WMT14 German–English dataset, the system with the best results is tarPOS, shown Table 3. The improvement brought by other methods is not significant, which may be related to the decoder's need for more accurate information. To predict the outcome. If it is necessary to make the translation results more in line with language rules, the target language knowledge may improve the model more than the source language knowledge. After all, if you already know the POS sequence of the target sentence or other language knowledge, it will be reduced to predict a certain extent. And the rules of the POS information of the source language and the POS information of the target language are quite different. In the English–Russian translation task, our method has a certain improvement compared with some methods. In the Russian–English translation task, compared with other methods, there is a 0.5 BLEU improvement. Our method also has a certain improvement in the resource-rich task compared with the base system.

Table 3. Experimental results on rich-resource translation tasks. The symbol—indicates that the loss during training is too low to train the model.

System	en2ru	ru2en	de2en
base [3]	29.65	30.17	35.47
Sennrich [8]	28.87	28.9	35.55
Niehues [22]	30.1	-	35.86
Yin [24]	29.01	28.95	35.58
Chakrabarty [16]	28.73	28.7	35.77
joint	22.68	-	35.53
srcPOS	29.86	29.7	35.45
tarPOS	29.78	29.44	35.87
srcPOS + tarPOS	30.21	30.82	35.66

5.5. Discussions

Due to the insufficient ability of the model to obtain features under low resources, the model's performance is greatly improved with the integration of POS information. For rich resources, since the model has more data to fit the natural distribution, the ability of the model to acquire features is better than that in the case of low resources. Even if part-of-speech information is added, the model will not improve with low resources. The degree of improvement is significant. Therefore, as the number of data increases, the model can acquire POS information or other language knowledge during training, which reduces the model's need for additional language knowledge. At the same time, if the POS information is excessively added, the model will contain a large number of POS features, which will reduce the features containing other information in the extracted features, thereby reducing the model's performance. We conducted related experiments for different data volumes to further verify the appeal conjecture. The experimental data are intercepted from the German–English translation task.

According to the experimental results in Table 4, the model accommodates POS information differently in different language pairs, which may be related to the POS and grammar between language pairs. The closer the language pairs are, the lower the model's ability to accommodate parts of speech. In this paper, different linear differences are performed on the loss function in srcPOS + tarPOS, so that the model loss function becomes line ($\mathcal{L} = 0.1 \times \mathcal{L}^{spos} + 0.5 \times \mathcal{L}^{spos'} + 0.1 \times \mathcal{L}^{KL} + \mathcal{L}^{word} + 0.3 \times \mathcal{L}^{tpos}$) and line2 ($\mathcal{L} = 0.2 \times \mathcal{L}^{spos} + 0.2 \times \mathcal{L}^{spos'} + 0.1 \times \mathcal{L}^{KL} + \mathcal{L}^{word} + 0.5 \times \mathcal{L}^{tpos}$). The results in Table 3 show that there is also some impact on the model by weighting the POS-related losses. This effect further proves that the model has a specific limit to the demand for POS information. The weights here are just randomly chosen. For optimal weights in different situations, relevant experiments are not carried out here.

Table 4. Experimental results of adding different degrees of POS information with different data volume.

System	0.2 m	0.5 m	1 m	4.5 m
base	18.07	21.71	25.1	35.47
joint	18.78	21.43	24.88	35.53
tarpos	18.35	22.46	24.95	35.87
srcpos + tarpos	18.82	22.33	24.92	35.66
line	19.4	22.26	25.3	35.76
line2	19.13	22.75	25.2	35.96

As mentioned earlier, POS information is already implicit in the text information. The features in the POS information are not contained in shorter dimensional features. The experimental results of the method proposed by Sennrich [8] are similar to the experimental results obtained on srcPOS on most language pairs. To compare whether these two methods are orthogonal, we merge the two methods and experiments on the English–Chinese translation task. According to the experimental results shown in Table 5, the two methods are orthogonal. It is also proved that the features fused utilizing loss function and direct concatenation have the same effect. And the fusion of these two methods reduces the results of the translation model, proving that there is a specific limit to the model's demand for POS features. At the same time, to compare when the integration of source data in the model structure is more conducive to improving model performance, we conducted further experiments on the srcPOS method.

Table 5. Comparison of two methods of fusing source POS Tags.

System	en2zh
base	19.33
joint	19.94
srcPOS	19.77
joint + srcPOS	19.47

In Table 6, the model loss function of the newly added srcPOS1 system is $\mathcal{L} = \mathcal{L}^{word} + \mathcal{L}^{spos}$; the model loss function of the srcPOS2 system is $\mathcal{L} = \mathcal{L}^{word} + \mathcal{L}^{spos'}$; and the model loss function of the srcPOS3 system is $\mathcal{L} = \mathcal{L}^{word} + \mathcal{L}^{spos} + \mathcal{L}^{spos'}$. The experimental results show that the two methods, srcPOS1 and srcPOS2, are beneficial to improve the performance of the translation model, and the fusion of these two methods is also improved compared with a single method. The results show that the two methods have different roles in the model structure. According to the model structure, srcPOS1 is more inclined to increase the attention of the translation model encoder to POS information. At the same time, srcPOS2 is separated from the encoder by a decoder. Its main impact should be on the decoder in the translation model. The simultaneous use of these two methods helps the model encoder and decoder to pay attention to the source POS information at the same time to improve the performance of the translation model.

Table 6. Comparison of methods for fusing POS tags at source.

System	zh2en	en2zh
base	21.25	19.33
srcPOS	21.86	19.79
srcPOS1	21.57	19.43
srcPOS2	21.73	19.44
srcPOS3	21.51	20.23

To better analyze the experimental results, we visualize the perplexity of the low-source translation task of the validation set during training on the Chinese–English and Chinese–Uighur translation task at different steps, as shown in Figure 4a–d. The perplexity of our proposed method is consistently lower than other methods. This shows that our method can generate more fluent sentences for NMT. Meanwhile, to show that our added loss has an effect on the model, we show the change in loss for the low-source translation task in Figure 5a–d. As can be seen from the figure, the translation loss of our method is also the lowest among all methods, which indicates that we guide the translation model to translate by adding POS loss, which is beneficial to reduce the loss of translation tasks and improve the performance of the translation model.

Although our method has been somewhat improved, the method still has some problems. For example, although the lexical annotation task is relatively simple compared with the translation task, the lexical annotation task cannot ensure that valuable information may be extracted accurately due to the sparse data. In addition, similar to the multitask approach, the loss function of the model has loss functions composed of multiple tasks, which leads to how the weights are defined before different loss functions. Although the linear interpolation can assign the weights of different loss functions, this does not entirely solve the problem, because reducing the influence of a particular loss function on the model by a weighting method inevitably leads to the model paying less attention to certain features. Therefore, setting the loss function for the primary and secondary relationship of the task is a crucial issue. And this paper does not address this part.

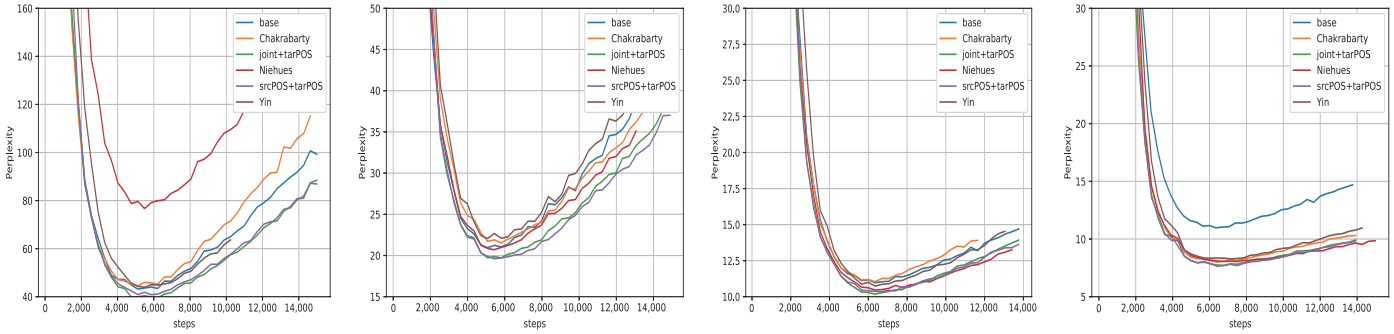

Figure 4. The perplexity of the validation dataset on different tasks: (**a**) The perplexity on English to Chinese translation task; (**b**) the perplexity on Chinese to English translation task; (**c**) the perplexity on Chinese to English translation task; and (**d**) the perplexity on Chinese to English translation task.

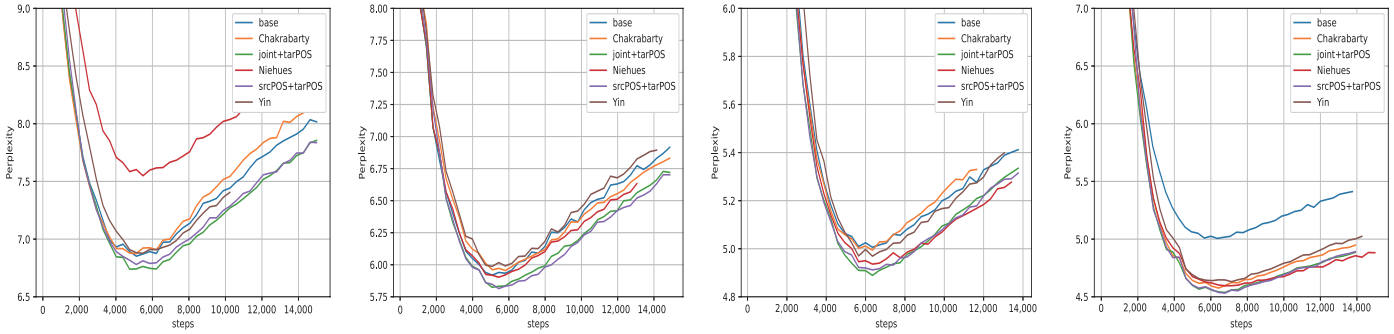

Figure 5. The loss of the validation dataset on different tasks: (**a**) The loss on English to Chinese translation task; (**b**) the loss on Chinese to English translation task; (**c**) the loss on Chinese to English translation task; and (**d**) the loss on Chinese to English translation task.

6. Conclusions

To better integrate POS information into the translation model, this paper proposes a method for integrating POS tags by combining a multitask approach and a reconstruction approach. The method achieves 23.88, 28.37, 23.00, and 20.46 for Chinese–Uighur and Chinese–English, respectively. The model improves the results by more than 1 BLEU compared with the baseline on these translation tasks. However, this paper only validates the method's performance with a limited number of translation tasks. The main validation is on Chinese-centric translation tasks and, therefore, on low-resource translation tasks that are also Chinese-centric. English, Chinese, and Uighur have their own linguistic and grammatical characteristics. Therefore, they can represent a large part of languages, which prevented us from extending the scope of the experiment. As for the translations between English and Uighur, there was no way to conduct experiments, because we did not have publicly available parallel data. Of course, there is still much room for improvement in this paper; specifically, (1) whether the method can be extended to other grammatical information and (2), since different grammatical information may contain some of the same information, information redundancy will lead to the degradation of model performance.

Author Contributions: Z.K. and N.Y. contributed to the conception of the study; Z.K. and N.Y. performed the experiment; Z.K. and N.Y. contributed significantly to the analysis and manuscript preparation; Z.K. performed the data analyses and wrote the manuscript; N.Y. and A.W. helped perform the analysis with constructive discussions. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 62166044).

Data Availability Statement: Some data are openly available in a public repository, and other data are available on request from the authors. We provide the link to these data.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

NMT	Neural Machine Translation
POS	Part of Speech
CCMT	China Conference on Machine Translation
IWSLT	The International Conference on Spoken Language Translation
WMT	Workshop on Machine Translation

References

1. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. In Proceedings of the 27th International Conference on Neural Information Processing Systems-Volume 2, Montreal, QC, Canada, 8–13 December 2014; pp. 3104–3112.
2. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
3. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6000–6010.
4. Wu, F.; Fan, A.; Baevski, A.; Dauphin, Y.; Auli, M. Pay Less Attention with Lightweight and Dynamic Convolutions. *arXiv* **2019**, arXiv:1901.10430.
5. Ranathunga, S.; Lee, E.S.A.; Prifti Skenduli, M.; Shekhar, R.; Alam, M.; Kaur, R. Neural Machine Translation for Low-resource Languages: A Survey. *ACM Comput. Surv.* **2023**, *55*, 229. [CrossRef]
6. Eriguchi, A.; Hashimoto, K.; Tsuruoka, Y. Tree-to-Sequence Attentional Neural Machine Translation. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 823–833.
7. Shi, X.; Padhi, I.; Knight, K. Does string-based neural MT learn source syntax? In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 1526–1534.
8. Sennrich, R.; Haddow, B. Linguistic Input Features Improve Neural Machine Translation. In Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers, Berlin, Germany, 7–12 August 2016; pp. 83–91.
9. Chen, H.; Huang, S.; Chiang, D.; Chen, J. Improved Neural Machine Translation with a Syntax-Aware Encoder and Decoder. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1936–1945.
10. Bastings, J.; Titov, I.; Aziz, W.; Marcheggiani, D.; Sima'an, K. Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 1957–1967.
11. Hashimoto, K.; Tsuruoka, Y. Neural Machine Translation with Source-Side Latent Graph Parsing. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 125–135.
12. Li, J.; Xiong, D.; Tu, Z.; Zhu, M.; Zhang, M.; Zhou, G. Modeling Source Syntax for Neural Machine Translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 688–697.
13. Wu, S.; Zhang, D.; Zhang, Z.; Yang, N.; Li, M.; Zhou, M. Dependency-to-dependency neural machine translation. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2018**, *26*, 2132–2141. [CrossRef]
14. Zhang, M.; Li, Z.; Fu, G.; Zhang, M. Syntax-Enhanced Neural Machine Translation with Syntax-Aware Word Representations. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 1151–1161.
15. Bugliarello, E.; Okazaki, N. Enhancing Machine Translation with Dependency-Aware Self-Attention. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 1618–1627.
16. Chakrabarty, A.; Dabre, R.; Ding, C.; Utiyama, M.; Sumita, E. Improving Low-Resource NMT through Relevance Based Linguistic Features Incorporation. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020; pp. 4263–4274.
17. Wu, G.; Tang, G.; Wang, Z.; Zhang, Z.; Wang, Z. An attention-based BiLSTM-CRF model for Chinese clinic named entity recognition. *IEEE Access* **2019**, *7*, 113942–113949. [CrossRef]

18. Labeau, M.; Löser, K.; Allauzen, A. Non-lexical neural architecture for fine-grained POS tagging. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 232–237.
19. Rei, M.; Crichton, G.K.; Pyysalo, S. Attending to characters in neural sequence labeling models. *arXiv* **2016**, arXiv:1611.04361.
20. Ren, S.; Zhou, L.; Liu, S.; Wei, F.; Zhou, M.; Ma, S. Semface: Pre-training encoder and decoder with a semantic interface for neural machine translation. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August 2021; pp. 4518–4527.
21. Caglayan, O.; Kuyu, M.; Amac, M.S.; Madhyastha, P.S.; Erdem, E.; Erdem, A.; Specia, L. Cross-lingual Visual Pre-training for Multimodal Machine Translation. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; pp. 1317–1324.
22. Niehues, J.; Cho, E. Exploiting Linguistic Resources for Neural Machine Translation Using Multi-task Learning. In Proceedings of the Second Conference on Machine Translation, Copenhagen, Denmark, 7–8 September 2017; pp. 80–89.
23. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
24. Yin, Y.; Su, J.; Wen, H.; Zeng, J.; Liu, Y.; Chen, Y. POS tag-enhanced coarse-to-fine attention for neural machine translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process. (TALLIP)* **2019**, *18*, 1–14. [CrossRef]
25. Wang, Y.; Zhai, C.; Hassan, H. Multi-task Learning for Multilingual Neural Machine Translation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 1022–1034.
26. Zhou, J.; Zhang, Z.; Zhao, H.; Zhang, S. LIMIT-BERT: Linguistics Informed Multi-Task BERT. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2020, Online, 16–20 November 2020; pp. 4450–4461.
27. Mao, Z.; Chu, C.; Kurohashi, S. Linguistically Driven Multi-Task Pre-Training for Low-Resource Neural Machine Translation. *Trans. Asian Low-Resour. Lang. Inf. Process.* **2022**, *21*, 1–29. [CrossRef]
28. Burlot, F.; Garcia-Martinez, M.; Barrault, L.; Bougares, F.; Yvon, F. Word representations in factored neural machine translation. In Proceedings of the Conference on Machine Translation, Copenhagen, Denmark, 7–11 September 2017; Volume 1, pp. 43–55.
29. Liu, Q.; Kusner, M.; Blunsom, P. Counterfactual data augmentation for neural machine translation. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 187–197.
30. Chen, G.; Chen, Y.; Wang, Y.; Li, V.O. Lexical-constraint-aware neural machine translation via data augmentation. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 11–17 July 2021; pp. 3587–3593.
31. Edunov, S.; Ott, M.; Auli, M.; Grangier, D. Understanding Back-Translation at Scale. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 489–500.
32. Tu, Z.; Liu, Y.; Shang, L.; Liu, X.; Li, H. Neural machine translation with reconstruction. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; Volume 31, pp. 3097–3103.
33. Gong, L.; Li, Y.; Guo, J.; Yu, Z.; Gao, S. Enhancing low-resource neural machine translation with syntax-graph guided self-attention. *Knowl.-Based Syst.* **2022**, *246*, 108615. [CrossRef]
34. Waldendorf, J.; Birch, A.; Hadow, B.; Barone, A.V.M. Improving translation of out of vocabulary words using bilingual lexicon induction in low-resource machine translation. In Proceedings of the 15th biennial conference of the Association for Machine Translation in the Americas (Volume 1: Research Track), Orlando, FL, USA, 12–16 September 2022; pp. 144–156.
35. Sethi, N.; Dev, A.; Bansal, P.; Sharma, D.K.; Gupta, D. Hybridization Based Machine Translations for Low-Resource Language with Language Divergence. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **2022**. [CrossRef]
36. Hlaing, Z.Z.; Thu, Y.K.; Supnithi, T.; Netisopakul, P. Improving neural machine translation with POS-tag features for low-resource language pairs. *Heliyon* **2022**, *8*, e10375. [CrossRef] [PubMed]
37. Maimaiti, M.; Liu, Y.; Luan, H.; Pan, Z.; Sun, M. Improving Data Augmentation for Low-Resource NMT Guided by POS-Tagging and Paraphrase Embedding. *Trans. Asian Low-Resour. Lang. Information Processing* **2021**, *20*, 1–21. [CrossRef]
38. Zheng, W.; Zhou, Y.; Liu, S.; Tian, J.; Yang, B.; Yin, L. A Deep Fusion Matching Network Semantic Reasoning Model. *Appl. Sci.* **2022**, *12*, 3416. [CrossRef]
39. Zheng, W.; Liu, X.; Yin, L. Sentence Representation Method Based on Multi-Layer Semantic Network. *Appl. Sci.* **2021**, *11*, 1316. [CrossRef]
40. Shi, X.; Yu, Z. Adding Visual Information to Improve Multimodal Machine Translation for Low-Resource Language. *Math. Probl. Eng.* **2022**, *2022*, 5483535. Available online: https://www.hindawi.com/journals/mpe/2022/5483535/ (accessed on 4 May 2022). [CrossRef]
41. Xu, M.; Hong, Y. Sub-word alignment is still useful: A vest-pocket method for enhancing low-resource machine translation. *arXiv* **2022**, arXiv:2205.04067.
42. Burchell, L.; Birch, A.; Heafield, K. Exploring diversity in back translation for low-resource machine translation. *arXiv* **2022**, arXiv:2206.00564.
43. Oncevay, A.; Rojas, K.D.R.; Sanchez, L.K.C.; Zariquiey, R. Revisiting Syllables in Language Modelling and their Application on Low-Resource Machine Translation. *arXiv* **2022**, arXiv:2210.02509.

44. Atrio, À.R.; Popescu-Belis, A. On the interaction of regularization factors in low-resource neural machine translation. In Proceedings of the 23rd Annual Conference of the European Association for Machine Translation, Ghent, Belgium, 1–3 June 2022.
45. Li, Z.; Sun, M. Punctuation as implicit annotations for Chinese word segmentation. *Comput. Linguist.* **2009**, *35*, 505–512. [CrossRef]
46. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Subword Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany, 7–12 August 2016; pp. 1715–1725.
47. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. Bleu: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
48. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
49. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
50. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
51. Freitag, M.; Al-Onaizan, Y. Beam Search Strategies for Neural Machine Translation. In Proceedings of the First Workshop on Neural Machine Translation, Vancouver, BC, Canada, 4 August 2017; pp. 56–60.