Research Article

Adding Visual Information to Improve Multimodal Machine Translation for Low-Resource Language

Xiayang Shi and Zhenqiang Yu

1College of Software Engineering, Zhengzhou University of Light Industry, Zhengzhou, China
2College of Mathematics and Information Science, Zhengzhou University of Light Industry, Zhengzhou, China

Correspondence should be addressed to Zhenqiang Yu; yzq2275stu@163.com

Received 5 May 2022; Revised 17 June 2022; Accepted 19 July 2022; Published 30 August 2022

Academic Editor: Muhammad Haroon Yousaf

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

Machine translation makes it easy for people to communicate across languages. Multimodal machine translation is also one of the important directions of research in machine translation, which uses feature information such as images and audio to assist translation models in obtaining higher quality target languages. However, in the vast majority of current research work has been conducted on the basis of commonly used corpora such as English, French, German, less research has been done on low-resource languages, and this has left the translation of low-resource languages relatively behind. This paper selects the English-Hindi and English-Hausa corpus, researched on low-resource language translation. The different models we use for image feature information extraction are fusion of image features with text information in the text encoding process of translation, using image features to provide additional information, and assisting the translation model for translation. Compared with text-only machine translation, the experimental results show that our method improves 3 BLEU in the English-Hindi dataset and improves 0.47 BLEU in the English-Hausa dataset. In addition, we also analyze the effect of image feature information extracted by different feature extraction models on the translation results. Different models pay different attention to each region of the image, and ResNet model is able to extract more feature information compared to VGG model, which is more effective for translation.

1. Introduction

Currently, machine translation is an important research direction. The main goal of information-driven machine translation is to decipher the source language information into the expected target language. The framework requires learning from sentence-aligned bilingual readiness information. Early statistical machine translation (SMT) is an information-driven model, which uses a probabilistic model to capture the interpretation process. The model relies on accepting a word as a basic element and a discriminant model based on maximum entropy and using the information obtained from the sentences [1].

Neural machine translation (NMT) models are usually two-part encoders and decoders; the work of the encoder is to encode a fixed-length language into a vector form according to certain rules, called a word encoding vector. The encoding vector hides the basic information of the source sentence. The text encoded by the encoder is sent to the decoder side, where the vocabulary of the target language is gradually deduced from the encoding vector, and finally the target discourse is obtained. Rely on a perfect decoder to produce complete sentences for the source language. Since the source and target languages usually have different lengths, the problem of gradient explosion between phrases is triggered by long short-term memory (LSTM) [2] and gated recursive units (GRU) [3]. Gated recursive unit (GRU) is a regular recurrent neural network. Sutskever et al. [4] proposed an encoder-decoder architecture for recurrent neural networks (RNNs).

The attention mechanism was originally proposed by [5] in their paper “Neural Machine Translation by Jointly Learning to Align and Translate” as an extension of work on the sequence to sequence model. The encoder-decoder
model only encodes a fixed-length sequence of sentences and decodes the output vector at each time step, and the attention was proposed to address the length limitation of the architecture. When the sequence of decoders is too long, the decoding results become worse. Attention was presented as a single approach for word alignment and translation. Word alignment is a prominent issue in machine translation, which analyzes the relationship between the input sequence and the output target sequence, and the translation model selects the most appropriate result for output based on exact relationship.

Multimodal machine translation (MMT) assists the system to translate higher quality target sequences by other information besides the text, such as images, audio, and video, which are incorporated into the translation model by different ways to assist the system. It has been shown that [6], by matching image information with text and performing training, produced better results than text-only translation.

For now, plain text-based machine translation with an encoder-decoder architecture is a widely used technique. This model accepts a sequence as input and encodes the information in the sequence into an intermediate representation. The decoder then decodes the intermediate representation into the target language. LSTM has been shown to be a good solution to the long-time sequence dependence problem. Sutskever et al. [4] proposed to encode the input sequence with an LSTM and decode that vector with another LSTM and obtain the output sequence. However, for long sentences, this encoder-decoder structure is unable to encode all the information. Bahdanau et al. [5] proposed attention mechanism and the problem was solved. In low-resource language translation, the model architecture based on attention mechanism has good performance. Bahdanau et al. [5] have good performance in WAT2021 with bidirectional RNN (BRNN) encoder and dual attention RNN decoder for multimodal machine translation.

In this paper, we investigate improving on the text-only machine translation model by adding image information as an auxiliary modality to improve the quality of the translated target sentences. In our experiments, we extract image features by different models and then add image features information to the encoder side of the baseline, fuse it with the initial word encoding information and position embedding information, and train an encoder containing image features, followed by a standard decoder module at the same time, to form a complete encoder and decoder model, thus improving the quality of English to Hindi translation. Comparing the experimental results, our approach improves 3 BLEU on the test set compared to the [7] method, and we also use different models to extract the image feature information and compare the impact of the extracted features on the translation results under different models.

2. Related Work

For English-Hindi translation tasks, the literature study found that the translation task still works on the basis of text-only [8–10]. Reference [8] used synthetic data, followed by the multimodal NMT setting [11], and obtained a BLEU score of 24.2 in Hindi-to-English translation. However, in the WAT2019 multimodal translation task English to Hindi, [12] obtained a higher BLEU score of 20.37 on the challenge test set. This score was later improved again in the WAT2020 [12] task, where a BLEU score of 33.57 was obtained on the challenge test set. In [7], a bidirectional RNN of encoder type (BRNN) and a double-attention RNN of decoder type with default settings are used, and pretrained words are used to embed the data of monolingual corpus and additional parallel corpus IITB. In the WAT2021 multimodal translation task, [13] attempted to employ phrase pairs to improve English to Hindi translation effectiveness and performance. Reference [14] proposed a reinforcement learning (RL) method to introduce sequence-level supervised signals as a systematic reward applied to the model when building the NMT system. Hausa, a Chadic language, is a member of the Afro-Asiatic language family. It is estimated that about 100 to 150 million people speak the language, with more than 80 million indigenous speakers. This is more than any of the other Chadic languages. Despite a large number of speakers, the Hausa language is considered low-resource in natural language processing (NLP) [15].

Reference [16] proposes to research the effect of inducing visual information in translation in datasets and suggests how to use data information for inducing text translation. Reference [17] proposed a Sequence Adaptive Memory (SAM) based cognitive translation model based on an improved version of the Cortical Learning Algorithm (CLA) with translation for English to Hindi. The model is capable of learning to create word pairs, dictionaries, and translation rules on very small data sets. Comparisons with traditional phrase-based approaches and current state-of-the-art methods are made and present comparable results.

There are scholars [18] who used pretrained fine-tuning for translation tasks between Hindi and English, and they used the pretrained mBART50 model [19] in a multitask setting, with translation as the primary task and self-supervised language modeling as a secondary task. The performance of the model is compared with the conventional fine-tuned mBART50 trained for the translation task. According to the current research results, we integrate image modal information with text information based on machine translation of text-only and input them into the neural network at the same time to increase the available information in the encoding stage. The encoder adopts the encoder module in transformer, which maps the input information into three matrices, which are query matrix Q, key matrix K, and value matrix V. The degree of influence of node i on node j is represented by the dot product of the query matrix Qi of node i and the key matrix Kj of node j, and after adjustment, it can become the weight of information of node j aggregating node i.

3. Method Implementation

3.1. Visual Feature Extraction. Matching images with text: In translation, the extracted image features and text information are fed into the translation model for fusion coding at the same time. For image features, we choose VGG11,
VGG19, and ResNet50, three models for feature information extraction to analyze whether the features extracted by different models contain rich visual information and their contribution to the translation task. We fine-tune the pre-trained VGG and ResNet models to the image feature information we need that is obtained.

3.2. Neural Machine Translation. Currently, most of the Neural Machine Translation (NMT) model is based on transformer [20], whose core is the attention mechanism. And it is used with multihead mechanism. Each attention header operates on an input sequence \( x = (x_1, \ldots, x_n) \) containing \( n \) elements, where \( x_j \in \mathbb{R}^d \), and computes a new sequence of the same length \( z = (z_1, \ldots, z_n) \), where \( z \in \mathbb{R}^d \):

\[
z_i = \sum_{j=1}^{n} y_{ij} (x_j W^V),
\]

where \( y \) is the weight coefficient, which is calculated by the softmax function:

\[
y_{ij} = \text{softmax} \left( \frac{(x_i W^Q)(x_j W^K)^T}{\sqrt{d}} \right),
\]

where \( W^Q, W^K, W^V \in \mathbb{R}^{d \times d} \) is the trainable parameter matrix for a particular layer.

In traditional machine translation (MT) tasks, the data are in the form of text-only, which is encoded and fed into the neural network, and the source utterance contains only textual information. As in Figure 1, we can divide the overall transformer process into several nodes at different times, such as word vector and location coding combination nodes, multiheaded attention nodes, and feedforward nodes.

If we want to integrate information from other modalities, we need to add node information from other modalities to the overall process, as an additional information to assisted system translation. Therefore, we extract feature information from the images and add image feature information as pseudo-words to the multimodal self-attentive layer.

3.3. Multimodal Self-Attention. [21] proposed a model with attention, which can better demonstrate the dependencies between each word in a sentence and called the self-attention mechanism, and the structural design is shown in Figure 1. In their model, the hidden state was calculated by a self-attentive and feedforward system, while positional encoding is used to show the regional relationships of each word in the sentence, adding a multiheaded attention mechanism to achieve parallelized processing. The processing of MT is accelerated due to the possibility of parallel processing, leading to better results than the model based on the benchmark recurrent system.

As mentioned earlier, in MMT, the extracted image feature information is only used as additional information to tutor the system for translation. Therefore, compared to text, visual feature information is not equally important. If the information is directly fused with the position embedding and word vector encoding at the initial location, unnecessary noise is generated. According to the multimodal self-attention mechanism proposed by [21], we add image feature information to the encoder part of the overall architecture, as shown in Figure 2.

Our work is an improvement of the encoding stage based on the transformer. The transformer is composed of several submodules; each one contains multiheaded attention, a feedforward neural network, and a residual structure in the submodule. Each submodule handles each token and its previous input relationship. In the beginning, the self-attentive mechanism is used to construct multiheaded attention modules. The approach is to apply self-attention to the same input several times, using separate normalization parameters. With the attention parameters set once, they can be reused several times to solve parallel processing problems.

In addition, the model easily learns the related information between different heads by computing the multihead attention method. During data processing, the self-attention module calculates and updates the weights according to the importance of token \( j \) in the sequence. As mentioned earlier, the input sequence is mapped into three matrices \((Q, K, V)\). The dimensionality of all three correlation matrices is \( \mathbb{R}^{d \times h} \), with \( d \) denoting the embedding dimension and \( h \) denoting the number of attention heads. The final output of each head will be integrated by a linear connection.

In the multimodal self-attention mechanism, with the attention parameters set once, they can be reused several times to solve parallel processing problems.
4.1. Dataset Analysis. This experiment uses the Hindi Visual Genome 1.1 dataset [22] (https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3267) and Hausa Visual Genome (https://github.com/hausanlp/HausaVisualGenome), which contains the training set (containing 28930 text sentences and 28928 images), the validation set (containing 988 text sentences and 998 images), the test set (1595 sentences and 1595 images), and the challenge set (1400 sentences and 1400 images). In the study, we found that some of the pictures in dataset do not have corresponding descriptive statements. There are also some data with duplicate statements. Therefore, we performed additional processing on the experimental data to remove the duplicate sentences in the text and the images without descriptive statements. At the same time, in order to guarantee the consistency of subsequent image feature extraction, we also eliminate the grayscale image and the corresponding descriptive statements from the dataset and finally collate the datasets for our experiments, 28,891 texts and images for each training set, 995 texts and images for each validation set, 1595 texts and images for test set, and 1595 texts and images for challenge test set, as shown in Tables 1 and 2.

<table>
<thead>
<tr>
<th>DataSet</th>
<th>TrainSet</th>
<th>ValidSet</th>
<th>TestSet</th>
<th>Challenge TestSet</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grayscale</td>
<td>36</td>
<td>3</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Repeat statement</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>No matching description</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Sentences/images</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 2: Hindi-English dataset processed results.

| Sentences/images | 28891 | 995 | 1595 | 1400 |

4.2. Image Feature Extraction. In this experiment, we used VGG11 and VGG19 [12] model and ResNet50 [23] model to extract image features information, respectively. The VGG19 model consists of 16 convolutional networks connected, followed immediately by 3 fully connected layers, each with a size of $3 \times 3$, followed by a $2 \times 2$ maximum pooling operation. And finally the data output from the fully connected layers is classified after softmax normalization. In our experiments, we modify the last two layers of the model as shown in Figure 3. The final softmax layer is discarded and the output of the fully connected layer is modified to match the data dimension of the experiment. The VGG11 model uses the same approach to extract image features.

Another feature extraction model used is ResNet50, through 4 blocks, with 3, 4, 6, and 3 bottlenecks in each block. In the latter blocks, normalization is achieved with softmax after average pooling and full connectivity. In the modifications similar to the VGG19 model, the image feature information is output by modifying the final output layer of the model. All extracted image feature data are saved in NPY data format, as shown in Figure 4.

4.3. Training. Our experiments are implemented based on the transformer in Fairseq [24] (https://github.com/pytorch/fairseq), an open source natural language processing tool released by Facebook, which is based on PyTorch
implementation. Our experiments are divided into two parts, which are text-only and multimodal comparison experiments for extracting image features by VGG11, VGG19, and ResNet50 model.

In our experiments we set the multihead attention head as 6 and the learning rate lr as 0.007; for the optimizer, we use Adam optimizer [25], where Adam $\beta_1$ and $\beta_2$ were 0.9 and 0.98, the training batch size is 64, and warmup updates are set to 8000 and trained on RTX3090 device with 24G memory.

Meanwhile, in the experiment, we set the parameter patience to 20 and continued training to 20 epochs when the current result of the model reached the best, and the whole training process was terminated when the best result of the model no longer improved. To avoid the error of the experiment, we evaluate the translation results for each model trained and select the best model result.

### 5. Result Analysis

We extracted image features by different models and performed multimodal machine translation experiments separately, and the experimental results were evaluated using BLEU [26]. We can find that the translation results under multimodality have a larger improvement compared to the text-only, with a 3 BLEU [https://github.com/moses-smt/mosesdecoder] improvement compared to [27] in Hindi-English and improved 0.47 BLEU. It can be found that the multimodal translation used in our experiments can translate the target language well, which indicates that extracting the image feature information corresponding to the text can effectively assist the system to translate and obtain more accurate target sentences, as shown in Table 3.

At the same time, we found that the image features extracted under different models also have different degrees of influence on the translation results. The image feature information extracted under the ResNet50 model is 0.6 BLEU higher than the image feature information extracted under the VGG19 model experimentally. Analyzing the model structure of both, VGG11, VGG19, and ResNet50 all use Convolution Neural Network (CNN), but the depth of the network used is different between them. VGG11 has 11 parametric network layers and 5 VGG blocks, where the first 2 blocks use a single convolutional layer and the next 3

<table>
<thead>
<tr>
<th>Model</th>
<th>Dataset</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only-txt NMT</td>
<td>Challenge</td>
<td>37.16</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>37.01</td>
</tr>
<tr>
<td>Multimodal NMT</td>
<td>Challenge</td>
<td>39.28</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>39.46</td>
</tr>
<tr>
<td>Only-txt transformer</td>
<td>Challenge</td>
<td>39.53</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>37.08</td>
</tr>
<tr>
<td>VGG11 MMT</td>
<td>Challenge</td>
<td>41.46</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>41.09</td>
</tr>
<tr>
<td>VGG19 MMT</td>
<td>Challenge</td>
<td>41.69</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>41.78</td>
</tr>
<tr>
<td>ResNet50 MMT</td>
<td>Challenge</td>
<td>42.29</td>
</tr>
<tr>
<td></td>
<td>Evaluation</td>
<td>42.70</td>
</tr>
</tbody>
</table>
blocks use a double convolutional layer. The first module contains 1 input channel and 64 output channels, and the output channels of each network immediately following are doubled, resulting in a final output of 512 dimensions. At the end of the network are three fully connected networks and a softmax structure, with 3136, 512, and 512 neural units in its fully connected layers.

VGG19 also consists of 5 VGG modules, but each module contains different convolutional layers. The convolutional layers of each VGG19 module are 2, 2, 4, 4, 3 plus the final fully connected and softmax layers, which constitute the entire VGG19 network structure. The VGG11 network only learns the contour features of the image and cannot clearly express the relationship built in each region. Compared with VGG11, the extra convolutional layer of VGG19 can learn the relationship between the parts of the image more accurately. For the ResNet50 network, there are 4 large blocks, each with 3, 4, 6, and 3 submodules, the network is initialized with individual convolution, and the feature information is output by full connectivity, resulting in a total of 50 layers. Although ResNet is deeper than VGG11 and VGG19, the weight parameter is smaller because the ResNet network uses global average pooling, which reduces the number of parameters and reduces overfitting and computation to some extent. Therefore, it is able to obtain image feature information better. It can be inferred that different network architectures pay different attention to the features of each part of the image and therefore extract different feature results, resulting in differences in the experimental results of the two models.

Figure 5 shows that different models pay different attention to each region of the image when extracting image features, and Figure 5(a) shows the feature extraction heatmap under the VGG19 model. We can find that, when mentioning the characteristics of the figure, more attention is paid to the figure of the character area “benches”; other areas are paid less attention. In contrast, the ResNet50 model is able to focus on some of the “benches,” which may account for the difference in translation results between the two auxiliary systems.

Figure 6 shows the heatmap of the relationship between source language sentences and target language sentence words for different feature extraction models, when the image features are extracted and the auxiliary model is trained on the dataset translation under the VGG19 model and the ResNet50 model. It can be found through the figure that there is no major difference between the two pictures; only in individual words, the degree of association is not the
same, which also thus shows that the fusion of the image feature information we extracted makes it possible to assist the model in translation. As mentioned earlier, when extracting features under different models, the focus on each region of the image is different, so the extracted feature information is not exactly the same, and therefore, the auxiliary information provided to the model is also different, resulting in different degrees of word-to-word association.

As Tables 4 and 5 show, with different models, for the probability of being selected for each word in the target language (results are taken in log), we can find that both models with added image features have better probability of selection for each word than the text-only model, because the image feature information, which provides additional auxiliary information to the model, increases the weight of the query matrix Q (mentioned in Chapter 4) in the model, allowing each target word to be found more accurately, which is the aspect where multimodal machine translation outperforms text-only.

Comparing with the sample translation result, as shown in Table 6, the translation result in the plain text is “a woman sitting on a metal bench,” which only expresses the basic meaning of the original sentence but does not accurately translate the color of the bench “blue.” The multimodal

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>a</th>
<th>Woman</th>
<th>Sitting</th>
<th>On</th>
<th>a</th>
<th>Metal</th>
<th>Blue</th>
<th>Bench</th>
</tr>
</thead>
<tbody>
<tr>
<td>Only-TXT</td>
<td>एक</td>
<td>-0.23</td>
<td>-0.10</td>
<td>-0.93</td>
<td>-0.14</td>
<td>-1.76</td>
<td>-0.07</td>
<td>-0.14</td>
<td>-0.13</td>
</tr>
<tr>
<td>VGG11 MMT</td>
<td>एक</td>
<td>-0.47</td>
<td>-0.24</td>
<td>-0.98</td>
<td>-0.13</td>
<td>-0.81</td>
<td>-0.14</td>
<td>-0.11</td>
<td>-0.13</td>
</tr>
<tr>
<td>VGG19 MMT</td>
<td>एक</td>
<td>-0.35</td>
<td>-0.23</td>
<td>-1.05</td>
<td>-0.12</td>
<td>-0.12</td>
<td>-0.12</td>
<td>-0.04</td>
<td>-0.14</td>
</tr>
<tr>
<td>ResNet50 MMT</td>
<td>एक</td>
<td>-0.42</td>
<td>-0.361</td>
<td>-1.26</td>
<td>-0.11</td>
<td>-0.18</td>
<td>-0.29</td>
<td>-0.10</td>
<td>-0.13</td>
</tr>
</tbody>
</table>

Figure 6: Different models extract image features and the degree of association between words. (a) VGG11 model. (b) VGG19 model. (c) ResNet50 model.

Table 4: The probability that each word is selected in Hindi dataset, the value of the log result.
translation result is "a woman sitting on a blue bench," which accurately translates the color "blue" of the bench in the source language due to the additional information provided by the image features. From the analysis of the above results, it is clear that the multimodal machine translation method we use can improve the quality of English-Hindi translation to some extent.

### 6. Conclusion

This experiment is based on the improved transformer architecture. Transformer has good realizations in machine translation of plain text. In recent years, in the field of multimodal machine translation research, transformer has also been adopted to improve the translation effect. Compared to models that use text-only information, multimodal machine translation is able to input image feature information matched with text as additional information into the encoder or decoder (this paper uses adding image feature information at the encoder side) on limited resources to assist the system in translation to help the system obtain a better target language. After the experiments, it can be seen that the model is designed to effectively use the information provided by images for translation, and the image feature information extracted by using different network models has different degrees of influence on the experimental results. In future work, we will also try adding image feature information to different models to improve the translation effect of low-resource languages.

### Data Availability

The code and data used to support the findings of this study are available from the corresponding author upon request.

### Conflicts of Interest

The authors declare that they have no conflicts of interest.

### Acknowledgments

This work was supported by the Transport Technology Project of Henan province (2021G5), the Foundation and Cutting-Edge Technologies Research Program of Henan Province (212102210540 and 212102210547), and Sichuan Science and Technology Program (no. 2021YFG0308).

### References


