ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10447334

VECTOR QUANTIZATION KNOWLEDGE TRANSFER FOR END-TO-END TEXT IMAGE MACHINE TRANSLATION

Cong Ma[1,2], Yaping Zhang[1,2], Yang Zhao[1,2], Yu Zhou[2,3], Chengqing Zong[1,2]*

[1]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[2]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[3]Fanyu AI Laboratory, Zhongke Fanyu Technology Co., Ltd, Beijing, China

ABSTRACT

End-to-end text image machine translation (TIMT) aims at translating source language embedded in images into target language without recognizing intermediate texts in images. However, the data scarcity of end-to-end TIMT task limits the translation performance. Existing research explores aligning continuous features from related tasks of text image recognition (TIR) or machine translation (MT) to alleviate the problem of data limitation, but the alignment in continuous vector space is extremely difficult and it inevitably introduces fitting errors resulting in significant performance degradation. To better align TIMT features with MT semantic features, we propose a novel Vector Quantization Knowledge Transfer (VQKT) method that employs a trainable codebook to quantize continuous features into discrete space. The quantization distribution of the MT feature is utilized as the teacher distribution to guide the TIMT model to generate similar discrete codes. Through alignment and knowledge transfer based on probability distribution, the TIMT model can better imitate the feature representation of the MT teacher model and generate high-quality target language translation. Extensive experiments demonstrate VQKT significantly outperforms the existing end-to-end TIMT performance.

Index Terms— Text image machine translation, vector quantization, quantization distribution, knowledge transfer.

1. INTRODUCTION

Text image machine translation (TIMT) task has been widely explored to translate the source language in images into the target language. Previous applications commonly use the cascade system, which extracts texts in images through the text image recognition (TIR) model and then translates it into target language with the machine translation (MT) model [1, 2]. However, cascade systems face the problems of error propagation, parameter redundancy, and long latency [3].

To overcome the shortcomings in cascade methods, researchers turn to exploring end-to-end methods [3], which utilizes an efficient encoder-decoder architecture to map the text image into the target language text directly. However, the vanilla end-to-end TIMT model faces the problems of data scarcity and the cross-modal cross-lingual transformation task is difficult to optimize. Although various techniques are applied to utilize external datasets from related tasks or transfer model knowledge of similar functions [4–12], all these methods are trying to transfer or align features in the continuous vector space, which inevitably introduces fitting errors due to the extreme difficulty for accurate alignment between continuous features as shown in Figure 1. To ease the difficulty in continuous space fitting, vector quantization methods have been widely studied to improve the capacity of hidden representation [13–20]. The multimodal codebook is designed for TIMT task [14], but the vanilla L2-Norm based image-text code alignment cannot fully transfer the quantization knowledge, causing the limited performance of end-to-end TIMT.

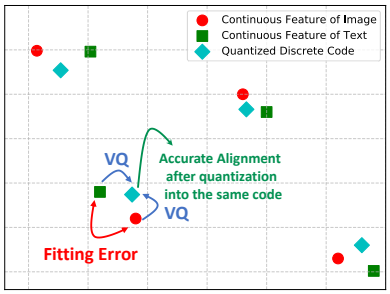

Fig. 1: Diagram of difference between continuous and discrete feature alignment. Discrete feature alignment is more accurate because the image and text features are quantized to the same discrete code which has no fitting error.

In this paper, we propose a novel Vector Quantization Knowledge Transfer (VQKT) method for end-to-end TIMT model. Different from L2-Norm based quantization alignment in MC-TIT [14], which only considers the alignment object of the single discretized code pair, quantization distribution alignment in our work can provide a probability distribution of input features across the entire codebook to provide more comprehensive alignment supervision. Specifically, features encoded by text image encoder are quantized into discrete codes and guided by the teacher distribution supervision of MT feature quantization during optimization. Thus, the

*Corresponding author.

This work has been supported by the National Natural Science Foundation of China (NSFC) under Grant No. 62006224 and 62106265.

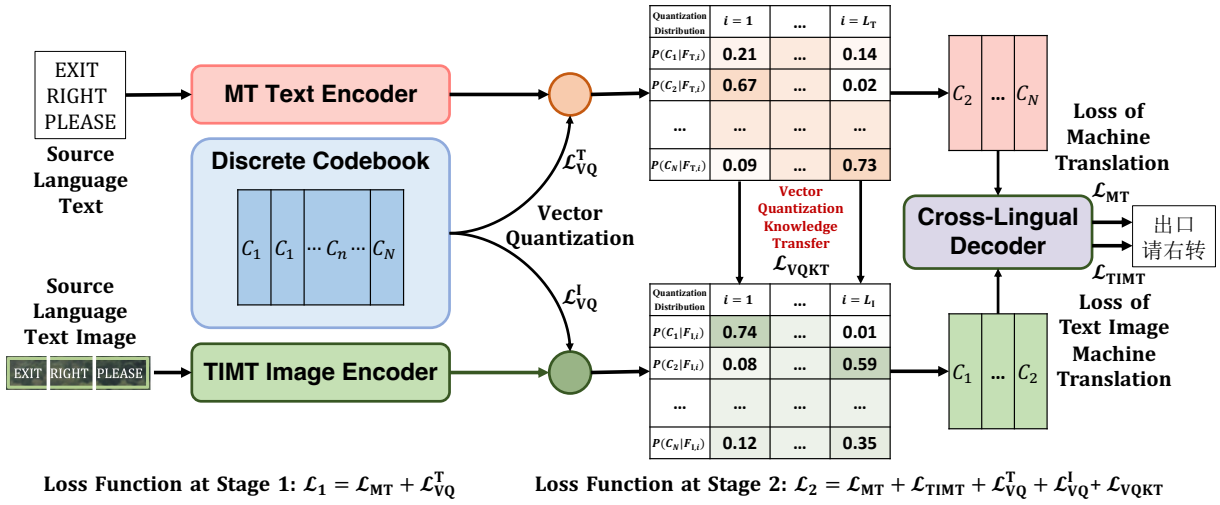

Fig. 2: Diagram of the proposed Vector Quantization Knowledge Transfer based Text Image Machine Translation Model.

TIMT discrete features can be more similar to the MT model and avoid fitting errors. Finally, the TIMT discrete features are fed into an MT decoder to generate high-quality translation results. The contributions of this paper are as follows:

- We propose a novel vector quantization knowledge transfer (VQKT) method for the end-to-end TIMT model, which effectively aligns the semantic features of the TIMT model with the MT model by alleviating the fitting errors in continuous feature space.
- Vector quantization object and quantization distribution alignment loss functions are jointly utilized to improve the effectiveness of knowledge transfer.
- Extensive experiments show that the VQKT method significantly outperforms existing end-to-end TIMT methods. Furthermore, the effectiveness of fuzing VQKT with other knowledge transfer methods shows the good generalization of our proposed method.

2. METHODOLOGY

As shown in Figure 2, our proposed vector quantization knowledge transfer method aligns the quantization distribution of TIMT and MT model to guide the TIMT encoder to obtain a similar discrete code sequence as the MT encoder.

2.1. Image Encoding

Let $I \in \mathbb{R}^{H\times W\times \mathrm{C}}$ be the text image containing source language in it, where H, W, and C represent height, width, and channel of text image respectively. The segmented patch sequence of text image is: $\mathbf{I}_P = \{I_{P,1}; I_{P,2}; \cdots; I_{P,L_I}\} \in \mathbb{R}^{d_P \times L_I}$, where $I_{P,i} \in \mathbb{R}^{d_P}$ denotes the patch image at i-th position and it resizes into patch vector of size $d_P = H \times C$. While $L_I = W/W_P$ means the length of the patch sequence and W_P denotes the width of the column patch. Then, the image features are encoded by a transformer encoder:

$$F_{\mathbf{I}} = \mathrm{TransformerEncoder}(\mathbf{W} \cdot \mathbf{I}_P + \mathbf{PE}) \quad (1)$$

where $F_{\mathbf{I}} \in \mathbb{R}^{d_I \times L_I}$ represents the encoded image feature sequence and d_I denotes the dimension of image features. $\mathbf{W} \in \mathbb{R}^{d_I \times d_P}$ represents a linear transformation matrix aims at mapping path image into patch embedding and $\mathbf{PE}$ denotes the sinusoidal position embedding as in [21].

2.2. Text Encoding

Assume $\mathbf{T} = \{T_1; T_2; \cdots; T_{L_T}\}$ is the source language texts embedded in image. T_i represents the i-th token in the sentence and L_T denotes the length of token sequence. A transformer encoder is utilized to extract text semantic features:

$$F_{\mathbf{T}} = \mathrm{TransformerEncoder}(\mathrm{Embedding}(\mathbf{X}) + \mathbf{PE}) \quad (2)$$

where $F_{\mathbf{T}} \in \mathbb{R}^{d_T \times L_T}$ represents the text feature sequence and d_T denotes the dimension of text features. Embedding$(\cdot)$ denotes the learnable matrix that maps the token index into token embedding. To align text features with image features through the same quantization codebook, the dimensions of text and image features are set the same: $d_T = d_I$.

2.3. Vector Quantization Knowledge Transfer

To better align image features with corresponding text features, vector quantization is utilized to transform continuous features into discrete space, and the quantization distribution alignment is used to guide the TIMT model to generate similar discrete codes as the MT model. Specifically, discrete codebook $\mathbf{C} = \{C_1; C_2; \cdots; C_N\} \in \mathbb{R}^{d_C \times N}$ contains N independent discrete code with the dimension of d_C. The quantization distribution refers to the probability distribution of input features quantized on the codebook $\mathbf{C}$:

$$P(C_j|F_i) = \frac{\exp(-\|F_i - C_j\|_2)}{\sum_n^N \exp(-\|F_i - C_n\|_2)}$$
$$Q_i = \arg\max_j P(C_j|F_i) \quad (3)$$

where $P(C_j|F_i)$ denotes the quantization probability of i-th continuous feature F_i and the final quantized code Q_i is the code with the highest quantization probability. C_j and C_n denote the j-th and n-th code in the codebook, respectively. By changing the conditional dependence of continuous feature F_i, the quantization probability of the image and text features are $P(C_j|F_{\mathbf{I},i})$ and $P(C_j|F_{\mathbf{T},i})$ respectively. To optimize the discrete codebook, the straight-through gradient estimation is utilized to copy gradients from the discrete results to continuous features as in [13]:

$$\mathcal{L}_{\mathrm{VQ}} = \|\mathrm{sg}(\mathbf{F}) - \mathbf{Q}\|_2^2 + \beta\|\mathbf{F} - \mathrm{sg}(\mathbf{Q})\|_2^2 \quad (4)$$

Architecture	Synthetic			Subtitle		Street
	En⇒Zh	En⇒De	Zh⇒En	En⇒Zh	Zh⇒En	Zh⇒En
ItNet [3]	18.43	15.71	11.38	16.91	10.07	0.94
CLTIR [4]	19.44	16.31	13.52	17.96	11.25	1.74
RTNet [8]	19.63	16.78	14.01	18.82	11.50	1.93
MTETIMT [5]	21.96	18.84	15.62	19.17	12.11	5.84
MHCMM [6]	22.08	18.97	15.66	19.24	12.12	5.87
E2E MC-TIT [14]	22.17	19.21	15.74	19.28	12.14	5.95
MTKD [7]	22.26	19.38	15.84	19.31	12.17	6.08
E2TIMT [9]	22.53	19.67	16.25	19.46	12.39	6.24
Our method: VQKT	**23.17**	**19.85**	**16.49**	**19.49**	**12.43**	**6.31**

Table 1: Comparison of end-to-end text image machine translation models.

where $\mathcal{L}_{\text{VQ}}$ represents the vector quantization loss function and $\text{sg}(\cdot)$ represents the stop-gradient operation. The first term in Eq.(4) is utilized to update discrete codes given the continuous features as ground truth, while the second term is a commitment loss as in [13] which aims to make sure the encoder commits to the discrete code space. $\mathbf{F}$ and $\mathbf{Q}$ denote continuous feature sequence and quantized code sequence respectively. Note that there are two versions of vector quantization loss $\mathcal{L}^{\mathbf{I}}_{\text{VQ}}$ and $\mathcal{L}^{\mathbf{T}}_{\text{VQ}}$ by changing the continuous features with image and text features as shown in Figure 2.

Through vector quantization, both image and text continuous features are mapped into discrete space, and the alignment of quantization probability is conducted to guide the TIMT encoder to generate similar discrete code as the MT encoder:

$$\mathcal{L}_{\text{VQKT}} = -\sum_{l=1}^{L}\sum_{n}^{N} P(C_n|F_{\mathbf{T},l}) \log P(C_n|F_{\mathbf{I},l}) \quad (5)$$

where $\mathcal{L}_{\text{VQKT}}$ denotes the vector quantization knowledge transfer loss function. The quantization distribution given text continuous features $P(C_n|F_{\mathbf{T},l})$ is utilized as the teacher distribution to guide the optimization of quantization distribution given image continuous features $P(C_n|F_{\mathbf{I},l})$. l denotes the l-th position in the feature sequence, while n denotes the n-th code in the discrete codebook. L and N represent the length of the feature sequence and the size of the codebook.

2.4. Cross-lingual Generation

Different from the vanilla translation model, the cross-lingual decoder in the VQKT method accepts the discrete feature sequence rather than the continuous feature sequence. The autoregressively generated target language decoder features are:

$$F_{\mathbf{D},t} = \text{TransformerDecoder}(\mathbf{Q}, F_{\mathbf{D},<t}) \quad (6)$$

where $F_{\mathbf{D},t}$ denotes the decoder feature at t-th generation step, and $F_{\mathbf{D},<t}$ represents the history of decoder features before step t. $\mathbf{Q}$ represents the quantized discrete code sequence. The decoder feature of TIMT ($F^{\mathbf{I}}_{\mathbf{D},t}$) and MT ($F^{\mathbf{T}}_{\mathbf{D},t}$) tasks are obtained by changing the conditional dependency of discrete code $\mathbf{Q}$ with $\mathbf{Q}^{\mathbf{I}}$ and $\mathbf{Q}^{\mathbf{T}}$. Thus the translation probability of TIMT and MT tasks are $P(\hat{y}^{\mathbf{I}}_t|\mathbf{I}, \hat{\mathbf{Y}}^{\mathbf{I}}_{<t})$ and $P(\hat{y}^{\mathbf{T}}_t|\mathbf{T}, \hat{\mathbf{Y}}^{\mathbf{T}}_{<t})$:

$$\begin{aligned} P(\hat{y}^{\mathbf{I}}_t|\mathbf{I}, \hat{\mathbf{Y}}^{\mathbf{I}}_{<t}) &= \text{softmax}(\mathbf{W}_o F^{\mathbf{I}}_{\mathbf{D},t}) \\ P(\hat{y}^{\mathbf{T}}_t|\mathbf{T}, \hat{\mathbf{Y}}^{\mathbf{T}}_{<t}) &= \text{softmax}(\mathbf{W}_o F^{\mathbf{T}}_{\mathbf{D},t}) \end{aligned} \quad (7)$$

where $\mathbf{W}_o \in \mathbb{R}^{|\mathcal{V}_{\mathbf{Y}}| \times d_D}$ represents the linear matrix, which is utilized to transform the decoder features into target language vocabulary space. $|\mathcal{V}_{\mathbf{Y}}|$ and d_D represent the size of the target language vocabulary and dimension of decoder features. Thus, the translation losses for TIMT and MT tasks are:

$$\begin{aligned} \mathcal{L}_{\text{TIMT}} &= -\sum_{t=1}^{L_{\mathbf{Y}}} \sum_{\hat{y}^{\mathbf{I}}_t \in \mathcal{V}_Y} \mathbb{I}(\hat{y}^{\mathbf{I}}_t = y_t) \log P(\hat{y}^{\mathbf{I}}_t|\mathbf{I}, \hat{\mathbf{Y}}^{\mathbf{I}}_{<t}) \\ \mathcal{L}_{\text{MT}} &= -\sum_{t=1}^{L_{\mathbf{Y}}} \sum_{\hat{y}^{\mathbf{T}}_t \in \mathcal{V}_Y} \mathbb{I}(\hat{y}^{\mathbf{T}}_t = y_t) \log P(\hat{y}^{\mathbf{T}}_t|\mathbf{T}, \hat{\mathbf{Y}}^{\mathbf{T}}_{<t}) \end{aligned} \quad (8)$$

where $\mathcal{V}_{\mathbf{Y}}$ denotes the target language vocabulary and y_t represents the translation ground truth at t-th position. $\mathbb{I}(\cdot)$ denotes the indicator function that takes the value of 1 when the decoded token by the TIMT or MT decoder is the same as ground truth and 0 otherwise.

2.5. Training and Inference

2.5.1. Training Process

To better control the training process and increase training stability, the training process is divided into two stages: (1) A trainable discrete codebook is incorporated in the middle of the MT encoder and decoder to achieve vector quantization based MT training. Thus, the machine translation loss and vector quantization loss functions are utilized in this stage: $\mathcal{L}_1 = \mathcal{L}_{\text{MT}} + \mathcal{L}^{\mathbf{T}}_{\text{VQ}}$. (2) Parameters of the discrete codebook, end-to-end TIMT model, and MT model are jointly optimized with vector quantization knowledge transfer loss to improve the translation performance of the TIMT model. Meanwhile, MT and TIMT losses are also kept to ensure the stability of the model training: $\mathcal{L}_2 = \mathcal{L}_{\text{MT}} + \mathcal{L}_{\text{TIMT}} + \mathcal{L}^{\mathbf{T}}_{\text{VQ}} + \mathcal{L}^{\mathbf{I}}_{\text{VQ}} + \mathcal{L}_{\text{VQKT}}$.

2.5.2. Inference Process

The inference process of the VQKT model just uses the TIMT image encoder, discrete codebook, and cross-lingual decoder to generate target language translation given the text image input. While MT encoder is abandoned because there are no recognized source language texts in the inference setting of end-to-end TIMT models.

3. EXPERIMENTS

3.1. Dataset

The public TIMT dataset released by [5] is utilized to train the VQKT model. The dataset consists of 1 million triple-

Architecture	BLEU	Δ
Multi-task based TIMT	22.87	-
w/ VQKT	23.95	↑ 1.08
Knowledge distillation based TIMT	24.13	-
w/ VQKT	25.88	↑ 1.75
Modal adapter based TIMT	24.63	-
w/ VQKT	26.29	↑ 1.66

Table 2: Generalization analysis of vector quantization knowledge transfer on English-to-Chinese validation set.

aligned training samples. Parallel text pairs are utilized in stage 1 to optimize the parameters of the MT model and discrete codebook, while the triple-aligned samples are utilized to train the end-to-end TIMT model and transfer knowledge from the MT model into the VQKT model. The evaluation sets have three translation directions: English-to-Chinese (En⇒Zh), English-to-German (En⇒De), and Chinese-to-English (Zh⇒En). Three evaluation domains are utilized in this dataset: synthetic, subtitle, and street-view domains. For more details of the dataset please refer to [5].

3.2. Experimental Setting

Transformer architecture [21, 22] is utilized as the backbone for TIMT and MT tasks. The batch size is 64, and the training steps for stages 1 and 2 are both 300,000. The codebook size is set to 3,072. Since the parameters of the codebook are only 1.29% of the total parameters, the optimization for the discrete codebook is quite efficient, which has little impact on the training time and memory consumption. Hyper-parameter β in $\mathcal{L}_{\text{VQ}}$ is set to 0.25 as in [13]. Trainable parameters are initialized with Xavier initiation method [23] and optimized with Adam optimizer [24] with $\beta_1 = 0.9, \beta_2 = 0.98$. The dropout rate is 0.1 and the initial learning rate is set to 2e-3. The whole training process is carried out on a single NVIDIA V100 GPU. SacreBLEU[1] is utilized as the metric to evaluate the translation performance [25, 26].

3.3. Comparison of TIMT performance

Table 1 shows the translation performance comparison between existing end-to-end TIMT methods and our proposed VQKT. To provide a fair comparison, the multimodal codebook proposed in [14] is reproduced with an end-to-end architecture (E2E MC-TIT). Different from L2-Norm based image-text alignment of mean-pooled discrete features [14], quantization distribution based VQKT in our work significantly improves the translation performance by 0.54 BLEU in average. Furthermore, VQKT outperforms existing best end-to-end methods (E2TIMT) with an average improvement of 0.35 points in the synthetic domain and an average improvement of 0.2 BLEU in all translation directions and domains, revealing the effectiveness of our method.

3.4. Generalization analysis of VQKT

By integrating VQKT with existing work as shown in Table 2, all end-to-end TIMT models are improved through discrete feature space alignment with an average improvement of 1.50 BLEU, indicating VQKT has good generalization by joint optimization of continuous and discrete feature alignment.

[1]https://github.com/mjpost/sacrebleu

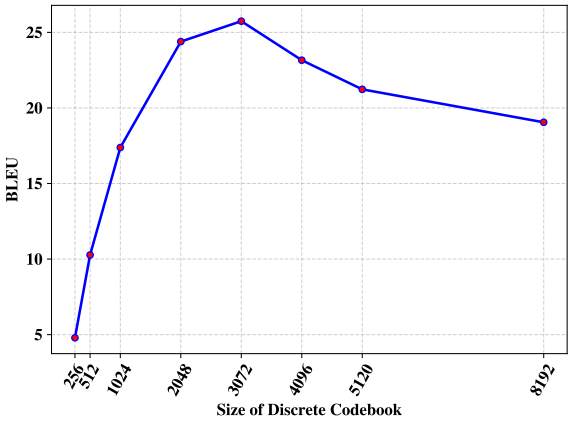

Fig. 3: Analysis of the size of discrete codebook on English-to-Chinese validation set.

Model	En⇒Zh	En⇒De	Zh⇒En
VQKT	**25.74**	**20.92**	**18.36**
w/o $\mathcal{L}_{\text{VQKT}}$	24.03	19.87	17.01
w/o $\mathcal{L}_{\text{VQKT}}$, $\mathcal{L}^{\text{I}}_{\text{VQ}}$	18.45	11.94	10.38
w/ random discrete code	3.25	3.02	2.99

Table 3: Ablation study on synthetic validation set.

3.5. Effect of the Size of Discrete Codebook

The size of the discrete codebook is a vital hyper-parameter for VQKT. We vary the setting of the size of the codebook as shown in Figure 3. When the size is 256, the representation capacity of the codebook is limited leading to severe performance degradation. As the size increases, the BLEU score gets better and the optimal size of the codebook is 3,072. When the size continues to increase, the performance is not further improved due to the increased redundant codes in a big discrete codebook.

3.6. Ablation Study

To evaluate the effectiveness of the VQKT and discrete codebook, we conduct an ablation study on the valid set as shown in Table 3. By removing VQKT loss, performance significantly decreases by 1.37 BLEU on average and further decreases 8.08 BLEU by removing VQ loss of image features. By replacing random discrete codes, the model collapses due to the random noise in the middle of the encoder and decoder, indicating the VQKT and discrete codebook are the key factors for improving translation performance.

4. CONCLUSION

In this paper, we propose a novel vector quantization knowledge transfer (VQKT) method designed for end-to-end TIMT. The key factor of our approach is to quantize the continuous feature sequence into discrete sequences and the knowledge transfer of quantization distribution. Experimental results reveal the effectiveness of VQKT. Furthermore, by jointly optimization with existing methods, VQKT achieves consistent improvements indicating our method has good generalization.

5. REFERENCES

[1] Ryota Hinami, Shonosuke Ishiwatari, Kazuhiko Yasuda, and Yusuke Matsui, "Towards fully automated manga translation," in *35th AAAI*, 2021.

[2] K. Chandra Shekar, Marilyn Cross, and Vignesh Vasudevan, "Optical character recognition and neural machine translation using deep learning techniques," *Innovations in Computer Science and Engineering*, 2021.

[3] Puneet Jain, Orhan Firat, Qi Ge, and Sihang Liang, "Image translation network," 2021.

[4] Zhuo Chen, Fei Yin, Xu-Yao Zhang, Qing Yang, and Cheng-Lin Liu, "Cross-lingual text image recognition via multi-task sequence to sequence learning," in *25th ICPR*, 2020, pp. 3122–3129.

[5] Cong Ma, Yaping Zhang, Mei Tu, Xu Han, Linghui Wu, Yang Zhao, and Yu Zhou., "Improving end-to-end text image translation from the auxiliary text translation task," in *26th ICPR*, 2022.

[6] Zhuo Chen, Fei Yin, Qing Yang, and Cheng-Lin Liu, "Cross-lingual text image recognition via multi-hierarchy cross-modal mimic," *IEEE TMM*, 2022.

[7] Cong Ma, Yaping Zhang, Mei Tu, Yang Zhao, Yu Zhou, and Chengqing Zong, "Multi-teacher knowledge distillation for end-to-end text image machine translation," in *17th ICDAR*, 2023.

[8] Tonghua Su, Shuchen Liu, and Shengjie Zhou, "Rtnet: An end-to-end method for handwritten text image translation," in *16th ICDAR*, 2021, pp. 99–113.

[9] Cong Ma, Yaping Zhang, Mei Tu, Yang Zhao, Yu Zhou, and Chengqing Zong, "E2timt: Efficient and effective modal adapter for text image machine translation," in *17th ICDAR*, 2023.

[10] Cong Ma, Xu Han, Linghui Wu, Yaping Zhang, Yang Zhao, Yu Zhou, and Chengqing Zong, "Modal contrastive learning based end-to-end text image machine translation," *IEEE/ACM TASLP*, 2023.

[11] Cong Ma, Yaping Zhang, Mei Tu, Yang Zhao, Yu Zhou, and Chengqing Zong, "CCIM: cross-modal cross-lingual interactive image translation," in *Findings of EMNLP*, 2023, pp. 4959–4965.

[12] Zhiyang Zhang, Yaping Zhang, Yupu Liang, Lu Xiang, Yang Zhao, Yu Zhou, and Chengqing Zong, "Layoutdit: Layout-aware end-to-end document image translation with multi-step conductive decoder," in *Findings of the EMNLP*, 2023, pp. 10043–10053.

[13] Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu, "Neural discrete representation learning," in *NeurIPS*, 2017, pp. 6306–6315.

[14] Zhibin Lan, Jiawei Yu, Xiang Li, Wen Zhang, Jian Luan, Bin Wang, Degen Huang, and Jinsong Su, "Exploring better text image translation with multimodal codebook," in *ACL*, 2023, pp. 3479–3491.

[15] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[16] Chae-Bin Im, Sang-Hoon Lee, Seung-Bin Kim, and Seong-Whan Lee, "EMOQ-TTS: emotion intensity quantization for fine-grained controllable emotional text-to-speech," in *ICASSP*. 2022, IEEE.

[17] Junhao Xu, Jianwei Yu, Xunying Liu, and Helen Meng, "Mixed precision DNN quantization for overlapped speech separation and recognition," in *ICASSP*. 2022, pp. 7297–7301, IEEE.

[18] Samir Sadok, Simon Leglaive, and Renaud Séguier, "A vector quantized masked autoencoder for speech emotion recognition," in *ICASSP*. 2023, IEEE.

[19] Yifei Wu, Chenda Li, and Yanmin Qian, "Light-weight visualvoice: Neural network quantization on audio visual speech separation," in *ICASSP*. 2023, IEEE.

[20] Haoyu Wang, Bei Liu, Yifei Wu, Zhengyang Chen, and Yanmin Qian, "Lowbit neural network quantization for speaker verification," in *ICASSP*. 2023, IEEE.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NeurIPS*, 2017, pp. 5998–6008.

[22] Yang Zhao, Jiajun Zhang, and Chengqing Zong, "Transformer: A general framework from machine translation to others," *Mach. Intell. Res.*, vol. 20, no. 4, pp. 514–538, 2023.

[23] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010, pp. 249–256.

[24] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.

[25] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *ACL*, 2002, pp. 311–318.

[26] Matt Post, "A call for clarity in reporting BLEU scores," in *WMT*, 2018, pp. 186–191.