TSCMR:Two-Stage Cross-Modal Retrieval

Zhihao Chen[1,3] and Hongya Wang[1,2,3](✉)

[1] School of Computer Science and Technology, Donghua University, Shanghai, China

[2] State Key Laboratory of Computer Architecture, Institute of Computing Technology, CAS, Beijing, China

[3] Shanghai Key Laboratory of Computer Software Evaluating and Testing, Shanghai, China

hywang@dhu.edu.cn

Abstract. Currently, large-scale vision and language models has significantly improved the performances of cross-modal retrieval tasks. However, large-scale models require a substantial amount of computing resources, so the execution of these models on devices with limited resources is challenging. Thus, it is paramount to reduce the model size and minimize computing costs of a model without sacrificing its performance. In this paper, we improved TERAN by dividing cross-modal retrieval into two stages: image-text coarse-grained matching and image-text fine-grained matching. Specifically, we present a novel approach called Two-Stage Cross-Modal Retrieval network(TSCMR). To reduce model size after model training, our approach utilized a new knowledge distillation method for Transformer-based models. Experiments have shown that our approach maintains a performance comparable to TERAN on the MS-COCO 1K test set, while being 2x smaller and 3.1x faster on inference.

Keywords: Cross-modal · Two-Stage Retrieval · Knowledge Distillation

1 Introduction

The rapid development of mobile internet has fueled an explosive growth in the volume of multimodal data comprised of images, text, and videos. Correspondingly, the demands from users with regard to data modalities have become increasingly diversified. Consequently, a significant shift towards cross-modal retrieval from single-modal retrieval has been observed in users' retrieval requests. For instance, corporations like Google have recently attempted to utilize textual descriptions to achieve cross-modal retrieval between text and images. The concept of cross-modal retrieval is aimed at promoting information interaction between different modalities, and as such, is focused on retrieving other modality samples with similar semantics through a modality sample. Given this aim, the presence of semantic relations between modalities becomes pivotal.

X. Yang et al. (Eds.): ADMA 2023, LNAI 14179, pp. 563–575, 2023.
https://doi.org/10.1007/978-3-031-46674-8_39

In recent years, the mainstream method for cross-modal retrieval has been to train large-scale pre-training models based on the Transformer [1] architecture to learn the semantic relationships between different modalities. These models can be divided into single-stream and dual-stream structures. However, to extract meaningful information from highly redundant datasets, complex models and a large amount of computational resources are required, regardless of the structure used. At present, many models have billions of parameters and demand more than 10GB of GPU memory for deployment, so it is difficult to efficiently execute them on resource-restricted devices. Furthermore, retrieving information using such models takes a long time. In light of these challenges, minimizing the storage and computation costs of the model while ensuring optimal performance is crucial.

TERAN utilizes the cosine similarity to generate the similarity score between each region and word, thus forming a region-word similarity matrix. By applying a pooling technique to the matrix, a global similarity score is obtained for the image and text. Notably, the computational time involved in calculating the similarity between an image and text is significantly higher than that of extracting the features for both. Constructing a matrix for a single image and text pairing is not time-consuming, but for a corpus of a hundred or more, the process becomes protracted.

To optimise the inference speed. This paper proposes a two-stage cross-modal retrieval model. Specifically, the two-stage cross-modal retrieval model divides the retrieval task into coarse-grained and fine-grained matching stages. In the first stage, global features representing images and text are added, and scores are derived from these features to identify top-performing candidates for the second stage. In the second stage, the model uses regional features of the images and word-level features of the text to calculate fine-grained similarity scores, which form the final basis for determining image-text similarity. By selecting top k scoring items from the coarse-grained phase, the model can also attain inference acceleration. Notably, this two-stage process is designed to reduce time and computational resource consumption during the fine-grained matching phase. After training, this paper use a discussion of a newly-developed Transformer distillation method to reduce model size.

2 Related Work

This section provides a comprehensive discussion of prior research on cross-modal retrieval through the use of joint image and text processing. The main architecture of this model, which is the Transformer Encoder architecture, was introduced. Furthermore, we elaborated on knowledge distillation and its implementation in models employing the Transformer Encoder architecture.

2.1 Joint Image and Text Processing for Cross-Modal Retrieval

At present, Transformer-based pretrained models are highly esteemed in both academia and industry for understanding visual and textual information due

to their excellent performance in cross-modal retrieval, attracting attention of researchers. These models are classified into two categories: single-stream structure models and dual-stream structure models, based on the current research.

The mainstream method for cross-modal retrieval is to train large-scale pretrained models based on the Transformer architecture to learn the semantic correspondence between different modalities. These models are divided into single-stream [2–4] and dual-stream [5–7] structures. Before inputting the model, image-text pairs require image and text feature extraction. Image features may be region features based on object detection [8], CNN-based global features or patch features like ViT [9] whereas text features usually follow the preprocessing method of BERT [10]. Single-stream structures combine text and image features, inputting them into a single Transformer block, and fusing multiple modality inputs through self-attention mechanisms. The final output value, identified by the cls token, determines the similarity of the inputted image-text pair. Single-stream structures learn cross-modal feature information more effectively, leading to better performance in the final evaluation metrics. Dual-stream structures input text and image features separately into two different Transformer blocks. One block processes image features, the other processes text features, and they each output the cls token representing the global feature for both image and text, respectively. Cosine similarity is then utilized to calculate the similarity between image-text pairs. However, the lack of interaction between image and text features diminishes accuracy. To solve this problem, some models include additional Transformer blocks within the dual-stream structure to achieve interaction between different modality features. Nevertheless, while performance improves, model complexity and parameters increase as well.

The TERAN [11] model proposed by Nicola et al. belongs to a dual-stream architecture that deals with cross-modal retrieval tasks via word-region alignment in image-text matching. The supervision is only employed at a global image-text level in this model. Fine-grained matching is implemented between the low-level components of images and texts, which includes matching of image regions and words to maintain the richness of information in both modalities. TERAN performs as well as single-stream models in image and text retrieval tasks. The fine-grained alignment method from TERAN provides new ideas for large-scale cross-modal information retrieval research.

2.2 Transformer Encoder

The model architecture we propose is mainly composed of Transformer [1] Encoder. Specifically, as shown in Fig. 1, the Transformer Encoder layer mainly includes two sub-layers: multi-head attention(MHA) layer and fully connected feed-forward neural network(FNN) layer.

The Multi-Head Attention (MHA) is constructed by combining multiple self-attention layers altogether. The objective of the attention layer is to gather information on the connection between each token and other tokens to determine their significance in the input sequence. We adopt three input vectors, namely,

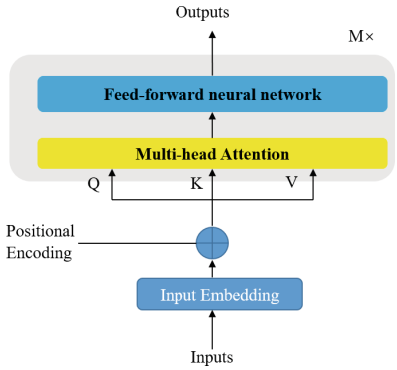

Fig. 1. Overview of Transformer Encoder.

the query(Q) vector, the key(K) vector, and the value(V) vector for our attention layer. The attention function can be expressed as the following formula:

$$\text{Attention}\,(Q, K, V) = \text{soft}\max\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

where d_k is the dimension of keys and acts as a scaling factor, and the factor $\sqrt{d_k}$ is used to mitigate the vanishing gradient problem of the softmax function in case the inner product assumes too large values.In essence, querying is akin to searching for information on a browser. The matching pages returned by the browser are keys, but what we require are the values that carry the desired information. By analyzing specific tokens and other tokens in a given sequence, we can determine their relevance and interdependencies with respect to another token. The self-attention mechanism involves multiple calculations, where different weight matrices are used for Q, K, and V, to facilitate this analysis.

The Transformer encoder incorporates a feedforward neural network layer, comprising of two linear transformation layers and a Rectified Linear Unit (ReLU) activation function, to acquire more comprehensive information.

2.3 Knowledge Distillation

Large-scale models are typically constructed using a single intricate network, or a composite of multiple networks. While these models demonstrate impressive performance and generalizability, small-scale models are often less expressive due to their smaller size. Knowledge distillation involves using knowledge gained from large-scale models to aid training of small-scale models, achieving comparable performance as large-scale models with reduced parameter size, thereby enabling model compression and acceleration.

Hinton et al. introduced the concept of "knowledge distillation" in [12]. The central idea is to improve the training of a small model by utilizing the knowledge learned by a large model. Therefore, the knowledge distillation framework

generally comprises a large model (known as the teacher model) and a small model (known as the student model). To enhance the quality of distilled knowledge and improve the performance of the student model, [13] proposed using an ensemble of models as the teacher model. [14] presented a knowledge distillation method based on the Transformer model structure that compresses and accelerates the pre-trained BERT model. Although it introduced a new loss function, [15] conducted experiments on the BERT model. In [16], a task-agnostic model compression method based on the BERT model was proposed.

In the field of natural language processing, the scale of pre-trained language models has been continuously expanding, and model compression has thus become increasingly important. To address this, [17] introduced a structured pruning method specifically designed for certain tasks called CoFi (Coarse and Fine-grained Pruning). The method combines pruning of coarse-grained units, such as self-attention layers and feedforward layers, with that of fine-grained units, such as heads and hidden dimensions. In addition, the authors proposed a hierarchical distillation method to dynamically learn the layer mapping relationship between the teacher and student models, which improves model performance. CoFi-compressed models achieve more than 10 times model acceleration, 95% parameter pruning, and maintain an accuracy rate of over 90% of the original model.

3 Method

In this section, we firstly introduce the model architecture. Then, we delineate the training objectives of the TSCMR. Lastly, we provide a comprehensive description of the knowledge distillation technique that was employed after completing the TSCMR training.

3.1 Model Architecture

Figure 2 displays TSCMR that includes the initial processing of images and text, an image encoder, a text encoder, and a method for calculating image and text similarity. Fast-Rcnn [8] is used for initial image processing and encodes input image I into an embedding sequence: $\{r_1, \cdots, r_n\}$. An image encoder consisting of four transformer encoders and one transformer encoder with two layers is used. The sequence is converted to $\{I_{cls}, r_1, \cdots, r_n\}$, where the token I_{cls} represents the global representation of the image, before inputting it into the image encoder. The text encoder adopts a combination of a 6-layer BERT model and one transformer encoder with two layers, converting input text T into an embedding sequence $\{T_{cls}, w_1, \cdots, w_n\}$. The token T_{cls} signifies the global representation of the text.

3.2 Training Objectives

TSCMR has two training objectives: image-text coarse-grained matching task (ITCG) and image-text fine-grained matching task (ITFG).

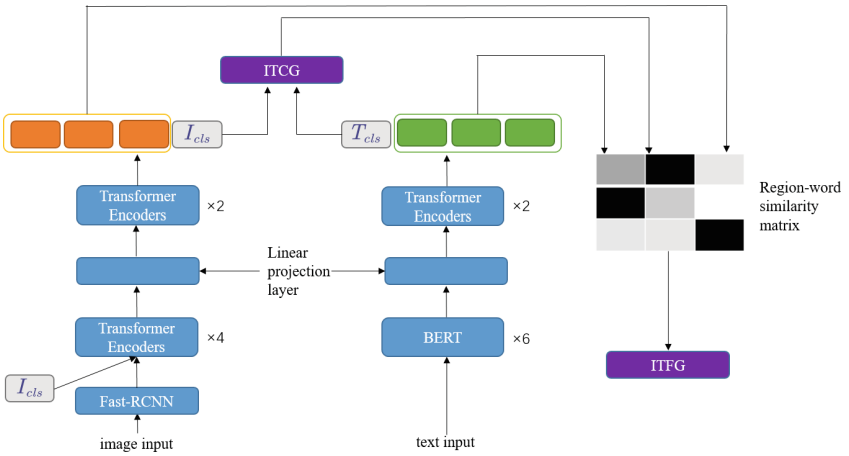

Fig. 2. The proposed TSCMR architecture. ITFG stands for image-text fine-grained matching, ITCG stands for image-text fine-grained matching. The orange boxes represents image region features and the green boxes represents word features. (Color figure online)

Image-Text Coarse-Grained Matching. In contrast to the TERAN, our new model architecture employs I_{cls} and T_{cls} for two-stage retrieval in order to reduce model inference time. After passing through the image and text encoders, we obtain the final image embedded sequence $\{I_{cls}, r_1, \cdots, r_n\}$ and text embedded sequence $\{T_{cls}, w_1, \cdots, w_n\}$. In the image-text coarse-grained matching stage, the S_{IT} similarity score is given by the cosine similarity between I_{cls} and T_{cls}, thus assigning higher scores to matched image and text pairs. The formula is as follows:

$$S_{IT} = \frac{I_{cls}^T T_{cls}}{\|I_{cls}\| \|T_{cls}\|} \tag{2}$$

After computing the coarse-grained similarity between image and text, we can employ the identical approach as described in [18] to compute the loss. This approach involves utilizing the hinge-based triplet ranking loss and directing attention towards hard negatives. The formula for calculating the loss is presented below:

$$L_{ITCG} = \max_{T'} [\alpha + S_{IT'} - S_{IT}]_+ + \max_{I'} [\alpha + S_{I'T} - S_{IT}]_+ \tag{3}$$

where $[x]_+ \equiv \max(0, x)$ and α is a margin that defines the minimum separation that should hold between the truly matching image-text pairs and the negative pairs, and calculates the negative examples T' and I' using the following method:

$$T' = arg \max_{z \neq T} S(z, T) \tag{4}$$

$$I' = arg \max_{y \neq I} S(y, I) \tag{5}$$

where (I,T) is a positive pair,z and y is negatives. During training, the dataset is divided into batches, thus negative examples are sampled from each batch.

Image-Text Fine-Grained Matching. At this stage, we drew upon the similarity matrix method employed in the TERAN, albeit abstaining from employing the I_{cls} and T_{cls} used in the previous phase. Cosine similarity is utilized to assess the similarity between the i-th region in I and the j-th word in T. Furthermore, the following approach is taken to compute the similarity matrix A:

$$A_{ij} = \frac{r_i^T w_j}{\|r_i\|\|w_j\|} \quad r_i \in I, w_j \in T \tag{6}$$

To calculate the global similarity between image and text, we used an appropriate pooling function to pool the similarity matrix. Inspired by [19,20], we adopted the max-sum pooling method, which selects the maximum value of each row in the similarity matrix A and sums them up. The specific formula is as follows:

$$S_{IT} = \sum_{w_j \in T} \max_{r_i \in I} A_{ij} \tag{7}$$

During this stage, we drew inspiration from the TopK algorithm. For each image I, we selected the finest K texts from the image-text coarse-grained matching scores to proceed to this stage. We calculated the fine-grained matching scores between I and the selected texts by employing a similarity matrix. Likewise, for each text T, we opt for the top M images with image-text coarse-grained matching scores, enter this stage, and calculate the fine-grained matching scores between T and these M images using similarity matrix. If the matching similarity scores of the text or image that genuinely matches are not in the top K or M sequence, we replace the lowest score with the newly found score. The hinge-based triplet ranking loss method is also implemented in this phase to calculate the loss, while the formula remains identical as follows:

$$L_{I2T-ITFG} = \max_{T'} [\alpha + S_{IT'} - S_{IT}]_+ \qquad T' \in K \tag{8}$$

$$L_{T2I-ITFG} = \max_{I'} [\alpha + S_{I'T} - S_{IT}]_+ \qquad I' \in M \tag{9}$$

The full training objective of two-stage retrieval model is:

$$L = L_{ITCG} + L_{I2T-ITFG} + L_{T2I-ITFG} \tag{10}$$

3.3 Distilling After Training

To minimize the model size, we utilized a Transformer-based knowledge distillation method to compress TSCMR. Drawing from [14], this work employs a hierarchical distillation technique to distill the multi-head self-attention modules, feedforward neural network modules, and embedding layers of every layer in the model, which is shown in Fig. 3.

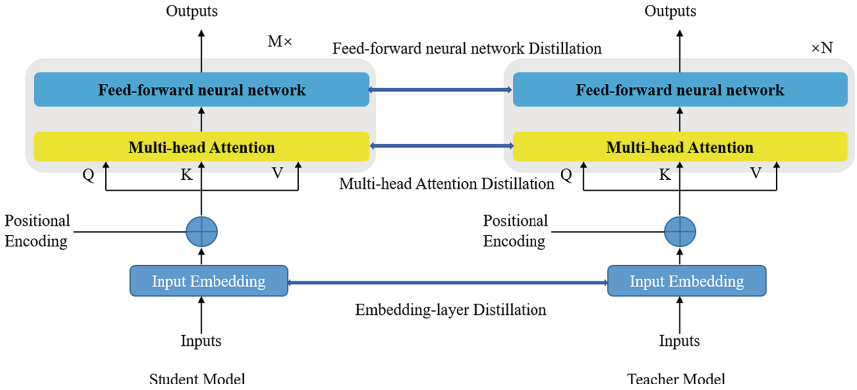

Fig. 3. The details of distillation

Embedding-Layer Distillation. The loss calculation for the embedding layer is as follows:

$$L_{embd} = MSE\left(E^S W_e, E^T\right) \tag{11}$$

where E^S and E^T respectively represent the embeddings of the student network and the teacher network. Since the embedding layer of the teacher network is usually smaller than that of the teacher model to reduce model size, the embedding of the student model is generally linearly transformed to project onto the space where the embedding of the teacher model is located. Finally, the mean squared error method is used to calculate the loss.

Transformer Encoder Distillation. We propose adopting the method of distillation every k layers for the Transformer encoder. Specifically, the loss is calculated every 3 layers when the teacher model consists of 12 layers while the student model has only 4 layers. Correspondingly, the first layer of the student model is aligned with the third layer of the teacher model, the second layer of the student model with the sixth layer of the teacher model, the third layer of the student model with the ninth layer of the teacher model, and the fourth layer of the student model with the twelfth layer of the teacher model. The loss of each Transformer encoder layer includes both the loss of the self-attention layer and the feedforward neural network layer.

The loss calculation of the self-attention layer follows the method below:

$$L_{attn} = \frac{1}{h}\sum_{i=1}^{h} MSE\left(A_i^S, A_i^T\right) \tag{12}$$

where h denotes the number of attention heads, A_i^S represents the attention score matrix of the i-th attention head in the student model, and A_i^T represents the attention score matrix of the i-th attention head in the teacher model.

The loss calculation method for the feedforward neural network layer is as follows:

$$L_{FFN} = MSE\left(H^S W_h, H^T\right) \tag{13}$$

where the matrices H^S and H^T refer to the hidden states of student and teacher networks respectively. Similar to embedding-layer distillation, the output of the student model is mapped to the same space as the output of the teacher network. This mapping enables the student model to learn from the teacher network and improve its performance.

Finally, by implementing the previously stated distillation objectives, we can calculate the overall distillation loss:

$$L = L_{embd} + L_{attn} + L_{FFN} \tag{14}$$

4 Experiments

This section introduces the datasets, evaluation metrics, and training process settings. The efficacy and efficiency of the cross-modal retrieval in TSCMR are evaluated. Moreover, we investigates the performance of TSCMR with the implementation of knowledge distillation in retrieval tasks, and the reduction in model size is also evaluated.

4.1 Datasets and Metric

This work employs two popular datasets, Microsoft COCO (MS-COCO) [21] and Flickr30K (F30K) [22], to train and test cross-modal retrieval tasks and investigate their effectiveness and efficiency. The MS-COCO dataset comprises 123,287 images, and each image has five corresponding texts. We utilize 113,287 images, 5,000 images, and 5,000 images for training, validation, and testing, respectively. The F30K dataset consists of 31,000 images, with five corresponding texts for each image. We select 29,000 images, 1,000 images, and 1,000 images for training, validation, and testing, respectively. For evaluation, this study uses Recall@K, a widely-used metric that precisely assesses the model's performance. The Recall@K value falls between 0 to 1 and indicates the proportion of appropriately identified positive samples in the model.

4.2 Settings

In the training of TSCMR, we use a image encoder consisting of a 4-layer transformer encoder and a 2-layer transformer encoder, and a text encoder consisting of a 6-layer BERT and a 2-layer transformer encoder. Image features and text features are projected into a common space of 1024 dimensions through a linear transformation for the final similarity calculation. In the experiment, we set the dropout rate to 0.1, use the Adam optimizer, set the epoch to 30, set the batch size of the MS-COCO dataset to 40, and set the batch size of F30K to 30. The learning rate is set to $1e{-}5$ during the first 20 epochs of training and $1e{-}6$ during

the remaining 10 epochs. When selecting the top-k images and texts with high similarity scores before entering the second stage, the k value is set to 15 for the MS-COCO dataset and 10 for the F30K dataset. After completing the training of TSCMR, we performed knowledge distillation. In the image encoder, we use a combination of 2-layer transformer encoder and 1-layer transformer encoder, while in the text encoder, we use a combination of 3-layer BERT and 1-layer transformer encoder. The dimensions and hyperparameters are kept unchanged during model training.

4.3 Results and Analysis

We compare our TSCMR method against the following baselines:VSRN [23],CAMERA [24],PFAN [25],MMCA [26],and TERAN. For the MS-COCO dataset, we present the result on the 1k test set. For 1k images, we computed the result through five-fold cross-validation on the 5k test set while averaging the obtained results.

Table 1. Results on the MS-COCO dataset,on the 1k test set

	Image Retrieval			Text Retrieval			
Model	R@1	R@5	R@10	R@1	R@5	R@10	SpeedUp
VSRN	62.8	89.7	95.1	76.2	94.8	98.2	-
CAMERA	63.4	90.9	95.8	77.5	96.3	98.8	-
PFAN	61.6	89.6	95.2	76.5	96.3	99	-
MMCA	61.6	89.8	95.2	74.8	95.6	97.7	-
TERAN	65	91.2	96.4	77.7	95.9	98.6	1.0x
TSCMR-100	63.6	90.1	95.6	75.2	95.1	98.5	6.7x
TSCMR-300	**64.8**	**91.1**	**96.7**	**77.2**	**95.6**	**98.8**	**3.1x**
TSCMR-500	64.9	91.3	96.8	77.4	95.8	98.9	1.9x

Table 1 reports the results on the MS-COCO dataset. The result reveals that the recall value of our method has experienced a significant downfall particularly in image retrieval with a drop of over a point in Recall@1, and over two points in text processing, when k is fixed to 100. Despite our model furnishing a 6.7 times higher retrieval speed compared to TERAN's method at k=100, our recall value suffered a huge setback. Nevertheless, when k is 300, the recall accuracy closely approximates that TERAN while maintaining a good balance between efficacy and viability. At $k = 500$, there is a minor improvement in recall value, however, the inference speed is only 1.9 times faster than TERAN.

Table 2 demonstrates that selecting at k of 100 results in a significant drop in the recall value, particularly for text retrieval, similar to the MS-COCO dataset. At k of 300 provides a well-balanced performance between recall value and efficiency that is not significantly different from TERAN. Increasing the value of k

Table 2. Results on the F30K dataset

	Image Retrieval			Text Retrieval			
Model	R@1	R@5	R@10	R@1	R@5	R@10	SpeedUp
VSRN	54.7	81.8	88.2	71.3	90.6	96	-
CAMERA	58.9	84.7	90.2	76.5	95.1	97.2	-
PFAN	50.4	78.7	86.1	70	91.8	95	-
MMCA	54.8	81.4	87.8	74.2	92.8	96.4	-
TERAN	59.5	84.9	90.6	75.8	93.2	96.7	1.0x
TSCMR-100	57.6	83.1	90.2	72.7	92.5	96.2	6.7x
TSCMR-300	**59.2**	**84.8**	**90.7**	**74.9**	**93**	**96.6**	**3.2x**
TSCMR-500	59.4	85	90.9	75	93.1	96.8	2x

to 500 does not substantially improve the recall value, but it significantly slows down the inference speed when compared to k set at 300.

During the testing phase, we made multiple selections of the optimal value of K for the MS-COCO and F30k datasets. Ultimately, we found that selecting a K value around 33 % of the size of the test set achieved an optimal balance between effectiveness and efficiency.

Table 3. Results on the MS-COCO dataset,on the 1k test set

	Image Retrieval			Text Retrieval			
Model	R@1	R@5	R@10	R@1	R@5	R@10	model size
TSCMR-300	64.8	91.1	96.7	77.2	95.6	98.8	100%
KL-TSCMR-300	64.1	91	96.5	76.2	94.8	98.3	50%

After the completion of the training phase for the two-stage retrieval model, knowledge distillation was conducted on the MS-COCO dataset. Table 3 of the report indicates that while the recall rate decreased slightly after the application of knowledge distillation, the size of the model reduced by 50%. Overall, this is a commendable achievement, especially for devices with GPU memory limitations.

5 Conclusions and Future Works

This paper proposes a new model architecture TSCMR for cross-modal retrieval, which is different from TERAN. The model consists of two stages: a image-text coarse-grained matching stage, based on global feature extraction, to filter irrelevant content before image-text fine-grained matching between word and image regions. Moreover, knowledge distillation is employed to reduce the model size after the training of the retrieval model. The experimental results demonstrate

that our model is capable of achieving outcomes comparable to those of TERAN on the MS-COCO 1K test set, with a 3.1x increase in inference speed and a 50% decrease in model size.

For the future work, the similarity calculation method has space for further improvement, and we plan to optimize it to enhance inference speed. We have currently tested our method on two datasets, and we intend to extend the testing to additional datasets in the future. To reduce model size, we will explore combining knowledge distillation, quantization, and pruning with our method.

Acknowledgments. The work reported in this paper is partially supported by NSF of Shanghai under grant number 22ZR1402000, the Fundamental Research Funds for the Central Universities under grant number 2232021A-08, State Key Laboratory of Computer Architecture (ICT,CAS) under Grant No. CARCHB 202118, Information Development Project of Shanghai Economic and Information Commission (202002009) and National Natural Science Foundation of China (No. 61906035).

References

1. Vaswani, A., Shazeer, N., Parmar, N., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, 30 (2017)
2. Li, L.H., Yatskar, M., Yin, D., et al.: Visualbert: a simple and performant baseline for vision and language. arXiv preprint arXiv:1908.03557 (2019)
3. Li, G., Duan, N., Fang, Y., et al.: Unicoder-vl: a universal encoder for vision and language by cross-modal pre-training. Proc. AAAI Conf. Artif. Intell. **34**(07), 11336–11344 (2020)
4. Qi, D., Su, L., Song, J., et al.: Imagebert: cross-modal pre-training with large-scale weak-supervised image-text data. arXiv preprint arXiv:2001.07966 (2020)
5. Lu, J., Batra, D., Parikh, D., et al.: Vilbert: pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In: Advances in Neural Information Processing Systems, 32 (2019)
6. Tan, H., Bansal, M., Lxmert: learning cross-modality encoder representations from transformers. arXiv preprint arXiv:1908.07490 (2019)
7. Huang, Z., Zeng, Z., Liu, B., et al.: Pixel-Bert: aligning image pixels with text by deep multi-modal transformers. arXiv preprint arXiv:2004.00849 (2020)
8. Ren, S., He, K., Girshick, R., et al.: Faster R-CNN: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, 28 (2015)
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
10. Devlin, J., Chang, M.W., Lee, K., et al.: Bert: pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
11. Messina, N., Amato, G., Esuli, A., et al.: Fine-grained visual textual alignment for cross-modal retrieval using transformer encoders. ACM Trans. Multimed. Comput. Commun. Appl. (TOMM) **17**(4), 1–23 (2021)
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
13. Freitag, M., Al-Onaizan, Y., Sankaran, B.: Ensemble distillation for neural machine translation. arXiv preprint arXiv:1702.01802 (2017)

14. Jiao, X., Yin, Y., Shang, L., et al.: Tinybert: distilling bert for natural language understanding. arXiv preprint arXiv:1909.10351 (2019)
15. Sanh, V., Debut, L., Chaumond, J., et al.: DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv preprint arXiv:1910.01108 (2019)
16. Sun, Z., Yu, H., Song, X., et al.: Mobilebert: a compact task-agnostic BERT for resource-limited devices. arXiv preprint arXiv:2004.02984 (2020)
17. Xia, M., et al.: Structured pruning learns compact and accurate models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1513–1528 (2022)
18. Faghri, F., Fleet, D.J., Kiros, J.R., Fidler, S.: VSE++: improving visual-semantic embeddings with hard negatives. In: BMVC. BMV A Press, 12 (2018)
19. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3128–3137 (2015)
20. Lee, K.H., Chen, X., Hua, G., Hu, H., He, X.: Stacked cross attention for image-text matching. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 201–216 (2018)
21. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
22. Plummer, B.A., Wang, L., Cervantes, C.M., Caicedo, J.C., Hockenmaier, J., Lazebnik, S.: Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In: ICCV (2015)
23. Li, K., Zhang, Y., Li, K., Li, Y., Fu, Y.: Visual semantic reasoning for image-text matching. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4654–4662 (2019)
24. Qu, L., Liu, M., Cao, D., Nie, L., Tian, Q.: Context-aware multi-view summarization network for image-text matching. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1047–1055 (2020)
25. Wang, Y., et al.: Position focused attention network for image-text matching. arXiv preprint arXiv:1907.09748 (2019)
26. Wei, X., Zhang, T., Li, Y., Zhang, Y., Wu, F.: Multi-modality cross attention network for image and sentence matching. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10941–10950 (2020)