Improving Cross-Modal Recipe Embeddings with Cross Decoder

Jing Yang Junwen Chen Keiji Yanai
The University of Electro-Communications, Tokyo, Japan
{yang-j,chen-j}@mm.inf.uec.ac.jp,yanai@cs.uec.ac.jp

ABSTRACT

In this paper, we propose an effective cross-modal embedding fusing decoder (Cross Decoder) for cross-modal recipe retrieval tasks. We introduce our Cross Decoder into a recent GAN and transformer-based method to improve the representation capability of the recipe embeddings. By reconstructing images through GAN using embeddings learned by our Cross Decoder, we increase the reliability of embeddings, as well as achieving high-quality image generation. In addition, with dynamic margins adopted in the retrieval loss, the performance of the whole framework is further improved. The experimental results show that our method outperforms the state-of-the-art methods on the Recipe1M dataset.

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**; • **Computing methodologies** → **Search methodologies**.

KEYWORDS

Cross-Modal Recipe Retrieval, Transformer

ACM Reference Format:
Jing Yang Junwen Chen Keiji Yanai. 2024. Improving Cross-Modal Recipe Embeddings with Cross Decoder. In *The Fifth Workshop on Intelligent Cross-Data Analysis and Retrieval (ICDAR '24), June 10–14, 2024, Phuket, Thailand.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3643488.3660303

1 INTRODUCTION

In recent years, the development of the Internet has made vast amounts of multimodal data available to people, and searching for cross-modal data has become an important issue. One of the typical tasks in cross-modal search is cross-modal recipe retrieval. The purpose of this task is to search for the corresponding recipe text for a given query image, or the corresponding recipe image for a given query recipe text. The challenge is that it is difficult to distinguish between similar recipe texts and meal images in recipe data. Recipe1M [5] proposed by Salvador *et al.* is a widely used dataset for cross-modal recipe retrieval. The recipe text has three sections: title, ingredients, and cooking instructions. At the same time, each recipe is accompanied by a recipe image. Recently, TNLBT [10] leverages advanced vision transformer [4] to encode recipe images, and trains the model with large batch size. With these advancements, TNLBT achieves state-of-the-art performance on the Recipe1M dataset. In this paper, we adopt TNLBT as the baseline model and further improve the performance. To summarize, our contributions are three-fold:

- We propose a Cross Decoder to improve the representation capability of the recipe embeddings by fusing the cross-modal recipe embeddings.
- We introduce dynamic margins into distance learning of TNLBT to adjust the learning difficulty.
- The results on the Recipe1M dataset show that our method outperforms the state-of-the-art methods.

2 RELATED WORK

From the beginning, a natural solution (Joint Embedding, JE) proposed by Salvador *et al.* [5] is to encode features from two different modalities into a joint embedding space while bringing the distributions of the encoded correspondences closer together to enable mutual search. Wang *et al.* introduces an efficient adversarial learning framework and proposed ACME [9], which achieves high retrieval accuracy by introducing a triplet loss with hard sample mining. Under the boom of Transformer [8] in natural language processing, H-T [6] introduces hierarchical transformers to encode recipe texts and a self-supervised learning strategy to explore complementary information between recipe texts. With the attention mechanism of Transformer, H-T outperforms ACME by a large margin. Then, T-Food [7] introduces the MultiModal Regularization (MMR) module on top of the H-T architecture to integrate the information between modalities and a variant of triplet loss with dynamic margins is proposed to adjust the learning difficulty of the model. To ensure the consistency of text information, several previous studies [9, 10] have generated images of text, but have ignored the relationship between images and text in the image generation process. The fusion of multimodal features has been shown to improve the accuracy of retrieval tasks [3]. ALBEF [3] has shown that using Cross Attention between images and text in image-text retrieval can improve the performance of both retrieval and pseudo text target generation. Inspired by ALBEF, we introduce our Cross Decoder into TNLBT to improve the representation capability of the recipe embeddings by considering the relationship between images and texts.

3 METHOD

The overview of our proposed method is shown in Fig. 1. We first introduce the recipe embedding encoding process in Sec. 3.1. Then, we describe the image and text embedding learning process in Sec. 3.2 and also introduce the implementation of our dynamic margin triplet loss. Finally, we introduce our Cross Decoder in Sec. 3.3.

ICDAR '24, June 10–14, 2024, Phuket, Thailand

ACM ISBN 979-8-4007-0549-6/24/06
https://doi.org/10.1145/3643488.3660303

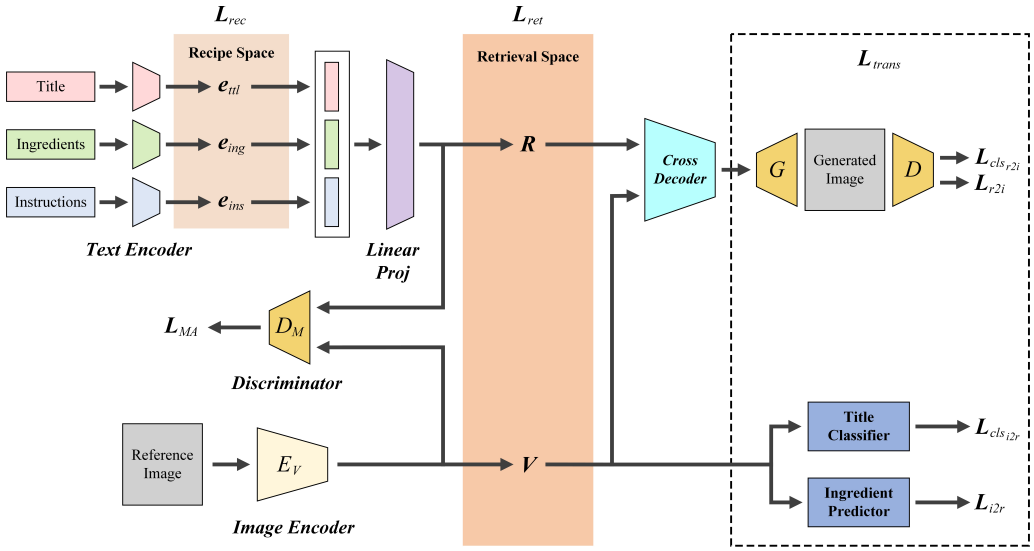

Figure 1: Overview of our proposed method. We build our method on top of TNLBT and introduce Cross Decoder to improve the representation capability of the recipe embeddings. First, the text encoder and image encoder are used to encode the recipe text and image, respectively. Then the encoded recipe and image embeddings R and V are aligned by adversarial learning and distance learning. Our Cross Decoder is used to fuse the cross-modal recipe embeddings, and GAN is used to generate images from the recipe embeddings.

3.1 Recipe Embedding Encoding

Following the architecture of TNLBT, as shown in Fig. 1 to learn more reliable recipe embeddings, first, three elements in the recipe text, title, ingredients, and instructions are encoded, respectively, with hierarchical transformer encoders $E_{ttl}, E_{ing}, E_{ins}$ proposed by H-T [6], and obtain embeddings $\boldsymbol{e}_{ttl}, \boldsymbol{e}_{ing}, \boldsymbol{e}_{ins}$ for each of the title, ingredients, and instructions. Then, these three embeddings are concatenated and encoded with the recipe text projection layer E_T to obtain the final recipe text embedding $\boldsymbol{R}$. As nearly 70% of the samples in Recipe1M [5] only have text information, according to H-T, a self-supervised loss function is used to explore the complementary meaning between the title and the ingredients and instructions in the recipe text. The self-supervised loss is used before encoding the title, ingredient, and instruction embeddings $\boldsymbol{e}_{ttl}, \boldsymbol{e}_{ing}, \boldsymbol{e}_{ins}$ into a single embedding $\boldsymbol{R}$. Specifically, a bi-directional triplet loss function is adopted, which is defined as follows:

$$\begin{aligned} L_{bi}^{'}(i,j) =& [c(\boldsymbol{e}_a^{(i)}, \boldsymbol{e}_b^{(j)}) - c(\boldsymbol{e}_a^{(i)}, \boldsymbol{e}_b^{(i)}) + \alpha]_+ \\ &+ [c(\boldsymbol{e}_b^{(i)}, \boldsymbol{e}_a^{(j)}) - c(\boldsymbol{e}_b^{(i)}, \boldsymbol{e}_a^{(i)}) + \alpha]_+ \end{aligned} \tag{1}$$

where, $\boldsymbol{e}_a$, $\boldsymbol{e}_b$ are different embedding sets, and the superscript (i) and (j) represent the index of the embedding set, $(\boldsymbol{e}^{(i)}, \boldsymbol{e}^{(i)})$ indicates a positive pair, and $(\boldsymbol{e}^{(i)}, \boldsymbol{e}^{(j)})$ indicates a negative pair. Also, $c(\cdot)$ is the cosine similarity, $[z]_+ = \max(0, z)$, and the margin α is set to 0.3. Following H-T [6], the equation (1) in a batch for any two of the title, ingredients, and instructions can be calculated as:

$$L_{bi}(a^{(i)}, b^{(i)}) = \frac{1}{B} \sum_{j=0}^{B} L_{bi}^{'}(i,j)\delta(i,j), \tag{2}$$

$$\delta(i,j) = \begin{cases} 0 & \text{if } i = j \\ 1, & \text{otherwise} \end{cases}$$

where B is the batch size, and δ is a correlation function, where $a, b \in \{ttl, ing, ins\}$. To prevent the problem of all embeddings becoming equal, as shown in Fig. 2, a linear layer is used to apply the mapping between embeddings and then calculate the loss function. $g_{a->b}(\cdot)$ is the mapping of a linear transformation from the space of set a to the space of set b. For example, $\boldsymbol{e}_{ttl}$ is mapped into the projection $\boldsymbol{e}_{ttl->ing}$ in the ingredient embedding space using $g_{ttl->ing}(\cdot)$, and then the loss with the ingredient embedding $\boldsymbol{e}_{ing}$ is calculated. The calculation of the loss function is as follows:

$$L_{rec}^{'}(ing, ttl) = L_{bi}(\boldsymbol{e}_{ttl}^{(i)}, \boldsymbol{e}_{ttl\to ing}^{(i)}) \tag{3}$$

For generalization, the loss function is defined as:

$$L_{rec}^{'}(a, b) = L_{bi}(\boldsymbol{e}_{a}^{(i)}, \boldsymbol{e}_{b\to a}^{(i)}) \tag{4}$$

Since there are three embedding spaces (title, ingredients, instructions), a total of six types of transformations, and six types of loss functions are calculated. Fig. 2 shows two of the six types of loss functions, which map the title embedding to other spaces. The loss function can be formulated as follows.

$$L_{rec} = \frac{1}{6} \sum_{a} \sum_{b} L_{bi}(a,b)\delta(a,b), \tag{5}$$

$$\delta(i,j) = \begin{cases} 0 & \text{if } i = j \\ 1, & \text{otherwise} \end{cases}$$

By using these three embeddings $\boldsymbol{e}_{ttl}, \boldsymbol{e}_{ing}, \boldsymbol{e}_{ins}$ learned in this way, more reliable text embeddings can be obtained, and the final recipe text embedding $\boldsymbol{R}$ is obtained through a linear projection layer E_T.

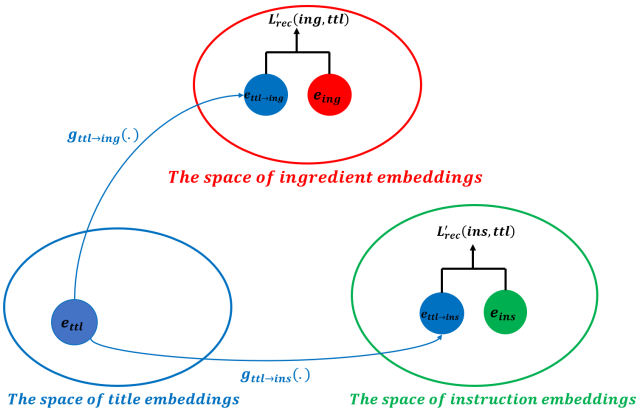

Figure 2: Visualization of mapping between title embeddings e_{ttl}, ingredients embeddings e_{ing}, and instruction embeddings e_{ins} and self-supervised learning loss $L'_{rec}(a, b)$.

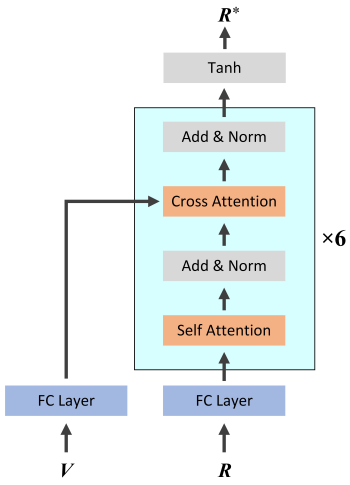

Figure 3: The architecture of Cross Decoder.

3.2 Image and Text Embedding Learning

Modality alignment. Similar to ACME [9], to alleviate the modality gap problem, adversarial learning is used to align the distributions of the image and text embeddings. Furthermore, in TNLBT, the distance learning is used to learn cross-modal embeddings of pairs. The purpose of alignment is to make the mapped image-text pair embeddings as close as possible. A discriminator D_M is used to align the embeddings from two modalities of text and image. This discriminator is to discriminate whether the given embedding is from an image or text. By learning the discriminator so that it cannot discriminate the origin of the given embedding, the loss function for aligning the embeddings from images and texts $(E_V(\boldsymbol{i}), E_T(\boldsymbol{t}))$ can be defined as in Equation (6). Following ACME, the loss function is defined as:

$$\begin{aligned} L_{MA} =& \mathbb{E}_{\boldsymbol{i}\sim p_{(i)}}[\log(D_M(E_V(\boldsymbol{i})))] \\ &+ \mathbb{E}_{\boldsymbol{t}\sim p_{(t)}}[\log(1 - D_M(E_T(\boldsymbol{t})))] \end{aligned} \tag{6}$$

The image encoder and recipe encoder maximize the loss function (6), and the discriminator D_M minimizes it. Through adversarial learning, the alignment of embeddings from the two modalities is more effectively performed.

Distance learning with dynamic margins. Given image and text embeddings $\boldsymbol{V} = E_V(\boldsymbol{i}), \boldsymbol{R} = E_T(\boldsymbol{t})$, to bring the embeddings of pairs closer and those of non-pairs farther apart. According to TNLBT [10], a triplet loss is used to reduce the distance between an anchor sample and a "positive" sample that is a pair, and to increase the distance from a "negative" sample that is not a pair, the Euclidean distance is used to calculate the similarity. Inspired by T-Food [7], we further develop the triplet loss by introducing a dynamic margin α_{dm}, which can be defined as:

$$\begin{aligned} L_{ret} =& \sum_{V}[d(\boldsymbol{V}_a, \boldsymbol{R}_p) - d(\boldsymbol{V}_a, \boldsymbol{R}_n) + \alpha_{dm}]_+ \\ &+ \sum_{R}[d(\boldsymbol{R}_a, \boldsymbol{V}_p) - d(\boldsymbol{R}_a, \boldsymbol{V}_n) + \alpha_{dm}]_+ \end{aligned} \tag{7}$$

where, $d(\cdot)$ is the Euclidean distance. The dynamic margin is set to 0.05 at the beginning, increases by 0.05 for each epoch, and is fixed when it reaches 0.3, which allows distance learning to be performed smoothly. Following ACME [9], the hard sample mining strategy is used to prioritize learning the hardest samples.

The consistency of embeddings. To ensure the consistency between the encoded embeddings and the original modality information, the same as ACME, TNLBT adopts two kinds of translation consistency losses. Specifically, L_{trans_r} is to verify the consistency of recipe embeddings by generating recipe images through the obtained recipe text embeddings, and L_{trans_i} is to verify the consistency of image embeddings by predicting the ingredients of the recipe through the obtained image embeddings. The loss that combines these two losses is defined as:

$$L_{trans} = L_{trans_r} + L_{trans_i} \tag{8}$$

However, different from TNLBT, we propose Cross Decoder to further increase this consistency by utilizing the attention mechanism between image and recipe embeddings. By using Cross Decoder, not only the accuracy of image generation from recipe embeddings but also the retrieval accuracy was improved.

Recipe Prediction from Image Embeddings. To ensure the consistency of the image embeddings, according to ACME, two kinds of recipe prediction loss are adopted. First, a multi-label ingredient predictor is constructed to predict the ingredients of the recipe using the image embedding. A total of 4102 types of ingredients are formulated as a one-hot vector. The loss is denoted as L_{i2r}. In addition, a classification loss $L_{cls_{i2r}}$ is used to ensure the image embedding is related to the correct category of the food image. The loss is denoted as $L_{cls_{i2r}}$. As a result, the loss to ensure the consistency of the final image embedding can be formulated as:

$$L_{trans_i} = L_{i2r} + L_{cls_{i2r}} \tag{9}$$

3.3 Cross Decoder

Fig. 3 shows the design of our Cross Decoder. The Cross Decoder consists of six attention blocks. The recipe text embedding $\boldsymbol{R}$ is fed into Cross Decoder *CrossDec* as a query, and the image embedding is input as key and value. The fused cross-modal recipe embedding

Table 1: Comparison with state-of-the-art methods on Recipe1M dataset.

	1k								10k							
	Image-to-Recipe				Recipe-to-Image				Image-to-Recipe				Recipe-to-Image			
	medR	R@1	R@5	R@10	medR	R@1	R@5	R@10	medR	R@1	R@5	R@10	medR	R@1	R@5	R@10
JE[5]	5.2	24.0	51.0	65.0	5.1	25.0	52.0	65.0	41.9	-	-	-	39.2	-	-	-
R2GAN[11]	2.0	39.1	71	81.7	2.0	40.6	72.6	83.3	13.9	13.5	33.5	44.9	12.6	14.2	35.0	46.8
ACME[9]	1.0	51.8	80.2	87.5	1.0	52.8	80.2	87.6	6.7	22.9	46.8	57.9	6.0	24.4	47.9	59.0
H-T[6]	1.0	60.0	87.6	92.9	1.0	60.3	87.6	93.2	4.0	27.9	56.4	68.1	4.0	28.3	56.5	68.1
X-MRS[2]	1.0	64.0	88.3	92.6	1.0	63.9	87.6	92.6	3.0	32.9	60.6	71.2	3.0	33	60.4	70.7
T-Food[7]	1.0	72.3	90.7	93.4	1.0	72.6	90.6	93.4	2.0	43.4	70.7	79.7	2.0	44.6	71.2	79.7
VLPCook[1]	1.0	73.6	90.5	93.3	1.0	74.7	90.7	93.2	2.0	45.3	72.4	80.8	2.0	46.4	73.1	80.9
TNLBT-C (baseline)	1.0	78.8	94.4	96.8	1.0	79.4	94.7	97.1	1.0	52.2	77.7	84.8	1.0	53.1	78.2	85.3
+CrossDec	1.0	80.9	95.4	97.6	1.0	80.8	95.5	97.8	1.0	55.5	80.2	87.0	1.0	54.5	79.5	86.6
+Dynamic margins	**1.0**	**81.8**	**95.9**	**97.8**	**1.0**	**81.2**	**96.0**	**97.9**	**1.0**	**56.5**	**81.0**	**87.6**	**1.0**	**55.7**	**80.2**	**87.1**

$\boldsymbol{R}^*$ containing rich relationships between images and texts obtained through the Fully Connected (FC) layer and Cross Decoder is as follows:

$$\boldsymbol{R}^* = CrossDec(FC(\boldsymbol{R}), FC(\boldsymbol{V})) \tag{10}$$

Then, the same as TNLBT, GAN is used to generate images from the recipe embedding (R^*). An adversarial loss L_{r2i} is used to make the generated images from the recipe embedding as close as possible to the real images. The loss can be formulated as:

$$\begin{aligned} L_{r2i} =& \mathbb{E}_{\boldsymbol{i}\sim p(\boldsymbol{i})}[\log(D_{r2i}(\boldsymbol{i}))] \\ &+ \mathbb{E}_{\boldsymbol{R}^*\sim p(\boldsymbol{R}^*)}[\log(1 - D_{r2i}(G(\boldsymbol{R}^*)))] \end{aligned} \tag{11}$$

The smaller this loss, the closer the generated images are expected to be to real food images. Similar to $L_{cls_{i2r}}$, to prevent the generated image from losing the original recipe text information, an additional classification loss $L_{cls_{r2i}}$ is adopted. A food-category classifier is used to classify the generated image, and the cross entropy loss is calculated as $L_{cls_{r2i}}$. Combining the two losses L_{r2i}, $L_{cls_{r2i}}$ above, the loss to ensure the consistency of the final recipe embedding is as follows:

$$L_{r2i} = L_{r2i} + L_{cls_{r2i}} \tag{12}$$

4 EXPERIMENTS

Dataset and Evaluation Metrics. The same as TNLBT and previous methods, we evaluate our method on the Recipe1M [5] dataset, which is split into 238,999, 51,119, and 51,303 image-recipe pairs for the training, validation, and test sets, respectively. The recipe information contains the title, ingredients, and instructions. Besides, for the self-supervised learning of recipes, the remaining 482,231 text-only samples are used. The same as previous research, we evaluate the retrieval accuracy based on the median rank (medR), Recall@{1,5,10} criteria, and the test size is divided into 1k and 10k. The reported retrieval accuracy is obtained by randomly selecting data from the test data 10 times and averaging the results.

Implementation Details. We adopt the TNLBT-C as the baseline model and use the same training settings. TNLBT-C leverages the CLIP-ViT [4] as the image encoder E_V. We train the model for 120 epochs with 8 NVIDIA A6000 GPUs, and the batch size is set to 768.

4.1 Comparison with State-of-the-Art Methods

We reproduce the baseline model TNLBT-C and evaluate the performance. Note that we found that there was a bug derived from the bug of ACME implementation [1] in the evaluation of the original TNLBT-C. After fixing the bug, the performance became lower than the results reported in [10]. As shown in Table 1, with our Cross Decoder, the Recall@{1} retrieval accuracy on Image-to-Recipe under 1k and 10k test size is improved by 2.7% and 6.3%, respectively. The Recall@{1} on Recipe-to-Image under 1k and 10k test size is improved by 1.8% and 2.6%, respectively. Then, by introducing dynamic margins in the triplet loss, the Recall@{1} on Image-to-Recipe under 1k and 10k test size is improved by 3.8% and 8.2%, respectively. The Recall@{1} on Recipe-to-Image under 1k and 10k test size is improved by 2.3% and 4.9%, respectively.

5 CONCLUSION

In this paper, we have explored the use of cross attention between images and texts for cross-modal recipe retrieval. We proposed a Cross Decoder to improve the representation capability of the cross-modal recipe embeddings and introduced dynamic margins in the triplet loss to improve representation learning. With the advantages of the Cross Decoder and dynamic margins, our method outperformed the existing state-of-the-art methods on the Recipe1M dataset.

Acknowledgments: This work was supported by JSPS KAKENHI Grant Numbers, 22H00540 and 22H00548.

REFERENCES

[1] Mustafa Shukor Nicolas Thome Matthieu Cord. 2023. Vision and Structured-Language Pretraining for Cross-Modal Food Retrieval. In *arXiv:2212.04267v2*.
[2] Ricardo Guerrero et al. 2021. Cross-modal Retrieval and Synthesis (X-MRS): Closing the modality gap in shared representation learning. In *ACMMM*.
[3] Junnan Li et al. 2021. Align before Fuse: Vision and Language Representation Learning with Momentum Distillation. In *NIPS*.
[4] Alec Radford et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *ICML*.
[5] Amaia Salvador et al. 2017. Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. In *CVPR*.
[6] Amaia Salvador et al. 2021. Revamping Cross-Modal Recipe Retrieval with Hierarchical Transformers and Self-supervised Learning. In *CVPR*.
[7] Mustafa Shukor et al. 2022. Transformer Decoders with MultiModal Regularization for Cross-Modal Food Retrieval. In *CVPR*.
[8] Ashish Vaswani et al. 2017. Attention is all you need. In *NIPS*.
[9] Hao Wang et al. 2019. Learning Cross-Modal Embeddings With Adversarial Networks for Cooking Recipes and Food Images. In *CVPR*.
[10] Jing Yang, Junwen Chen, and Keiji Yanai. 2023. Transformer-Based Cross-Modal Recipe Embeddings with Large Batch Training. In *MMM*.
[11] Bin Zhu et al. 2019. R2GAN: Cross-Modal Recipe Retrieval With Generative Adversarial Network. In *CVPR*.

[1] The detail of the bug on ACME was reported in X-MRS [2].