Transformer Decoders with MultiModal Regularization for Cross-Modal Food Retrieval

Mustafa Shukor\(^1\)* Guillaume Couairon\(^1,2\) Asya Grechka\(^1,3\) Matthieu Cord\(^1,4\)

\(^1\)Sorbonne University  \(^2\)Meta AI  \(^3\)Meero  \(^4\)Valeo.ai

Abstract

Cross-modal image-recipe retrieval has gained significant attention in recent years. Most work focuses on improving cross-modal embeddings using unimodal encoders, that allow for efficient retrieval in large-scale databases, leaving aside cross-attention between modalities which is more computationally expensive. We propose a new retrieval framework, T-Food (Transformer Decoders with MultiModal Regularization for Cross-Modal Food Retrieval) that exploits the interaction between modalities in a novel regularization scheme, while using only unimodal encoders at test time for efficient retrieval. We also capture the intra-dependencies between recipe entities with a dedicated recipe encoder, and propose new variants of triplet losses with dynamic margins that adapt to the difficulty of the task. Finally, we leverage the power of the recent Vision and Language Pretraining (VLP) models such as CLIP for the image encoder. Our approach outperforms existing approaches by a large margin on the Recipe1M dataset. Specifically, we achieve absolute improvements of 8.1\% (72.6 R@1) and +10.9\% (44.6 R@1) on the 1k and 10k test sets respectively. The code is available here: https://github.com/mshukor/TFood.

1. Introduction

Multimodal learning, especially Vision and Language Pretraining (VLP), has become an attractive research field with many applications such as cross-modal retrieval [45], Visual Question Answering (VQA) [1] and visual reasoning [63]. An interesting playground for multimodal learning is computational cooking, which encompasses different tasks such as food categorization [40], food perception [43], recommendation [18], or retrieval [48].

In this work, we focus on recipe-image retrieval, which consists in retrieving the image corresponding to a given recipe and vice versa. This task has gained much attention in the recent years since the release of Recipe1M [48], a dataset containing one million textual recipes and their corresponding images. The textual recipe is composed of 3 entities: title, ingredients, and instructions.

Most previous works [6, 19, 48, 59, 64] rely on specific (dual) encoders for images and text recipes that embed the two modalities in a shared latent space. The recipe encoder usually considers the recipe entities as independent and encodes them in separate modules [58]. These dual encoders are trained using pairwise, triplet or any contrastive loss optimization to align the visual and textual representations.

We propose a novel strategy for cross-modal recipe retrieval based on a new architecture and learning framework. Since recipe entities (title, ingredients, instructions) are highly correlated, we propose a hierarchical transformer that explicitly leverages the intra and inter entity dependencies. For the visual encoder, we consider vision transformers such as ViT, but also recent models such as CLIP-ViT pretrained on large multimodal datasets. We expect the latter to be more robust to the noise in the Recipe1M dataset.
which has been scraped from cooking websites. For efficient large-scale retrieval, we utilize dual encoders, but we consider a more complex image-recipe interaction during the training, which we materialize by a novel transformer based module with Image-Text Matching loss that acts as a regularization to better align the encoders representation. We also propose a new adaptive triplet loss with a dynamic margin which changes according to the difficulty of the task. The whole architecture and learning scheme is presented on Fig 1. Our contributions are two-fold:

- Deep architecture design: (a) we propose a new recipe encoder with transformer decoders that captures the interactions between recipe entities, (b) and we leverage VLP models trained on large-scale datasets for the image encoder. (c) We complete this architecture with a multimodal block that consists of inter-connected transformers, especially designed for the training. We keep only the unimodal encoders for efficient cross-modal retrieval at test time.

- Training framework: (a) We propose a novel Multi-Modal Regularization that consists of an Image-Text Matching loss on top of the multimodal module. (b) In addition, we introduce a new adaptive triplet loss with dynamic margin that adapts to the difficulty of the task.

We demonstrate the interest of our approach with extensive evaluation and comparison on the Recipe1M dataset.

The work is organised as follows; we detail some of the related work in section 2. We then present our approach in section 3. The method is validated in the experiments section 4. Finally, the work is summarized in section 5.

2. Related Work

2.1. Multi Modal Learning

Vision and Language Pretraining (VLP) has been successful for many downstream tasks such as visual question answering (VQA), image captioning, NLRVR\(^2\) and cross modal retrieval [10, 24, 30, 41]. They are mainly based on dual encoders (DE) [26, 46], multimodal encoders (ME) [11, 28, 35, 36, 72] or a combination of both [25, 38, 51, 53, 68]. Dual encoders, such as CLIP [46] and ALIGN [26], use separate encoders for each modality, and a contrastive loss to align the modalities. These models are efficient during inference, however, they are trained on massive amounts of data and rely on global similarity without exploiting the fine-grained modalities interaction.

Multimodal encoders, on the other hand, employ a heavy transformer that takes the tokens of both modalities and rely on several pretraining tasks such as Masked Language Modeling (MLM) [13], Image-Text Matching (ITM) [10, 38], Masked Image Modeling [2, 51] or prefixLM [12, 61]. Despite their success in vision-language tasks, they suffer from computational complexity due to self and cross attention between modalities, limiting their adaptability for large scale tasks. To make these models more efficient during inference, some work adopts re-ranking of top k examples after a first ranking using the dual encoders [30, 41]. Despite the success of these approaches for cross-modal retrieval, the efficiency of the model at large scale is still a bottleneck.

2.2. Contrastive Learning

Contrastive learning is a wide area in representation learning, especially self supervised learning (SSL) [5, 8, 9, 22], where the model is forced to be invariant to input transformations. This is done by encoding the positive pairs (e.g. the image and its transformed version) close in the latent space and the negative pairs (e.g. different images) far away, however, while contrastive learning is becoming mostly associated with SSL, the idea goes back to distance metric learning [67]. Interestingly, the contrastive loss is not applied to the underlying representation, but after projecting it into another space. This is similar to what we propose, although, we differ in; (a) we have 2 projection modules, one linear and another more complex, (b) we work in a multimodal setting. While most of the SSL work use the InfoNCE loss [44], here we follow other work on cross modal retrieval and use the triplet loss. Triplet loss [14, 49, 62] takes an anchor, positive and negative example, with an ob-

<table>
<thead>
<tr>
<th>Title query</th>
<th>Ingredient query</th>
<th>Instruction query</th>
<th>GT</th>
<th>Top 5 retrieved images</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mint Chocolate Chip Frosting</td>
<td>1 cup Unsalted Butter, 2 tablespoons Heavy Cream, 2 drops Green Food Coloring, ...</td>
<td>Add sugar, cream, peppermint, and food coloring...</td>
<td>Source: Chocolate Cupcakes with Mint Chocolate Chip Frosting</td>
<td><img src="image1.png" alt="Image" /> <img src="image2.png" alt="Image" /> <img src="image3.png" alt="Image" /> <img src="image4.png" alt="Image" /> <img src="image5.png" alt="Image" /></td>
</tr>
<tr>
<td>Honey-Grilled Chicken</td>
<td>1 broiler-fryer chicken, halved, 34 cup butter, melted, 14 cup honey, ...</td>
<td>Place the halved chicken in a large, shallow container. ...</td>
<td>Grill chicken, skin side up.</td>
<td><img src="image6.png" alt="Image" /> <img src="image7.png" alt="Image" /> <img src="image8.png" alt="Image" /> <img src="image9.png" alt="Image" /> <img src="image10.png" alt="Image" /></td>
</tr>
<tr>
<td>The Best Kale Ever</td>
<td>1/2 cup Kale, 1/4 teaspoon Olive Oil, 1/4 teaspoon Red Pepper Flakes.</td>
<td>Wash and cut kale off the stems...</td>
<td>Heat olive oil on medium heat and add garlic ...</td>
<td><img src="image11.png" alt="Image" /> <img src="image12.png" alt="Image" /> <img src="image13.png" alt="Image" /> <img src="image14.png" alt="Image" /> <img src="image15.png" alt="Image" /></td>
</tr>
</tbody>
</table>
objective of making the difference between the positive and negative distances are larger than a predefined margin $\alpha$. Several improvements have been selected to be the best triplets, such as cross batch mining [60], semi hard negatives [49], batch hard triplets [23] or truncated triplet [56]. However, little work have investigated the choice of the margin; in [70] the authors propose multi stage training where in each stage the margin is increased. In the fashion domain, [71] propose adaptive margin based on the semantic distance between product descriptions.

### 2.3. Cross Modal Recipe/Image Retrieval

The task consists of retrieving the recipe corresponding to a dish image and vice versa, and it is based on computing the similarity between image and recipe embeddings in a shared latent space. Many datasets have been proposed for food computing [4, 7, 17, 21, 27, 29, 39, 48]. Recipe1M and Recipe1M+ [39, 48] are the largest image/recipe datasets that consist of 800 k/13 M images respectively and 1 M recipes, structured in 3 entities; title, ingredients and instructions. It contains also the class of the recipe (e.g., pizza, green beans...). Despite the large number of images in Recipe1M+, Recipe1M is still only used for comparison.

Most Cross modal Image/Recipe retrieval approaches encode the recipe and the image using separate encoders [6, 16, 19, 39, 48, 58, 59]. The first methods rely on recurrent networks (e.g., LSTMs [6, 48]) to encode the ingredients and instructions, which are embedded using classical approaches such word2vec [42] and skip-thoughts [29]. On the image side, many work use classical image encoders such as ResNet-50 [6, 16, 48, 58, 73].

Due to this challenging task, some work add discriminative or generative regularizations. Discriminative regularizations try to leverage the category or the class of the recipe in the shared latent space to obtain more semantic embeddings [6, 39, 52, 59]. Generative regularizations include adversarial losses to align the distributions of the 2 modalities [58, 64, 69] or use Generative Adversarial Networks (GANs) to reconstruct the images from either the recipe or image embedding [52, 73], Though adversarial based approaches show good results, the discriminative ones remain much simpler and more efficient to train.

Ingredients and instructions do not contribute equally in the alignment, some ingredients can be seen clearly in the image while others does not appear. This motivates some work to include attention modules in the recipe encoder [19, 33, 59] or the image encoder [19] to weight differently several tokens/regions when fusing the 2 modalities. Seeing the success of image transformers in vision [15, 37, 54] and text [13], some work started recently to use transformers, showing promising results [20, 47].

Despite being successful for image and language tasks, multimodal transformers are still little investigated in the cooking context. Few work have tried to use cross attention modules between the 2 modalities [33, 34], or a multimodal transformer encoder [31], but their performance lag behind the SOTA work.

### 3. T-Food framework

Given a dataset composed of image and recipe pairs, we propose to align the representations of the two modalities for cross-modal retrieval. Fig. 1 illustrates the approach. First, each modality is encoded separately into two sequences of token embeddings. Notably, the recipe encoder fuses the information of title, ingredients and instructions into a single sequence. These two sequences are used in two different manners: (1) we extract two fixed-size embeddings that are used for retrieval at test time, and are brought closer together with a triplet loss at train time. (2) we feed the two sequences into the MultiModal regularization (MMR) module, which computes a fine-grained alignment score (Image-Text Matching loss) between the input image and recipe.

#### 3.1. Dual Encoders (DE)

In this section, we detail the image and recipe encoders (i.e., Dual Encoders (DE)).

**Image Encoder** Inspired by the success of vision transformers [15, 37, 54] and particularly for the underlying task
of cross-modal retrieval [47], we propose to use a vision transformer (i.e., ViT B/16) as an image encoder. In addition, fine-tuning large vision/language models on downstream tasks has been shown to be an effective strategy [10, 50], thus we propose to fine-tune the CLIP ViT B/16 [46] which was trained on a large-scale dataset scraped from the internet. Specifically, the image encoder $E_I$ encodes an input image $I$ into a sequence of image tokens $t_I = E_I(I)$ along with a [CLS] token output that is linearly projected to form a fixed-size image embedding $e_I$.

**Recipe Encoder** Different from other works that use recurrent networks and pretrained embeddings, we follow [47] using hierarchical transformers to encode the recipes as raw text. As illustrated in Fig. 2, each tokenized sentence in the list of ingredients ($t_{ins}^1$), instructions ($t_{ins}^2$) and the title $t_{ttl}^1$ is processed by a transformer $T$. For the ingredients and instructions, another transformer $HT$ processes the output of the first one (i.e. the sentence embeddings), in a similar way to obtain the embeddings $t_{ins}^2 = HT(T(t_{ins}^1))$ and $t_{ins}^3 = HT(T(t_{ins}^2))$ for instructions and ingredients respectively. For the title, the embedding is obtained from the first transformer $t_{ttl}^3 = T(t_{ttl}^1)$.

However, we argue that encoding each recipe entity separately, as done in [47], is not effective to capture the interactions between them. To this end, we propose to add another level of hierarchy to capture the interactions between all element tokens that leads to better intra-fusion. Specifically, a transformer decoder with self and cross attention $HTD(Q, K, V)$ takes the output tokens of each entity as a query $Q$ and the concatenation of the tokens of the other 2 remaining entities as keys $K$ and values $V$, the final tokens are obtained by concatenating all the tokens:

$$
t_{ttl}^3 = HTD(t_{ttl}^1; [t_{ttl}^2, t_{ins}^2]; [t_{ins}^2, t_{ins}^2]),
$$
$$
t_{ins}^3 = HTD(t_{ins}^1; [t_{ttl}^3, t_{ins}^2]; [t_{ttl}^2, t_{ins}^2]),
$$
$$
t_{ins}^3 = HTD(t_{ins}^2; [t_{ttl}^3, t_{ttl}^2]; [t_{ttl}^2, t_{ins}^2]),
$$
$$
t_R = [t_R^1; t_R^2; t_R^3; t_{ins}^1].$$

Finally, the output tokens of each recipe entity are averaged, concatenated and linearly projected to obtain the final recipe embedding $e_R$.

### 3.2. MultiModal Regularization (MMR)

Inspired by other work on vision and language pretraining [10, 30, 51], we propose a MultiModal Regularization (Fig. 3) that consists of transformer decoders [55] with an Image-Text Matching loss (ITM) [10] to further guide the dual encoders during the alignment process. Different from other work that use multimodal encoders during training and testing, here our multimodal modules are used only during training. Compared to other representation learning approaches, such as self-supervised learning, [8] where a simple projection layer is added on top of the feature representation to compute the contrastive loss, here, this projection layer is replaced by a more complicated multimodal module, and the contrastive loss is replaced by ITM.

The main block of MMR consists of a transformer decoder which particularly uses cross attention: the Queries ($Q$) come from one modality while the Keys ($K$) and Values ($V$) come from the other modality (we do not distinguish between $K$ and $V$, which denoted as $KV$). In other words, the queries are updated by taking into account the dependencies between the 2 modalities. Specifically, each vector of $Q$ is updated by taking a weighted sum of the vectors of $KV$ of the other modality, where each weight is the similarity of each one of these KV vectors to the underlying $Q$ vector. Our choice for using transformer decoder is based on their computation efficiency compared to that of a transformer encoder, which takes all the tokens as $Q$, $V$ and $K$.

In addition, we choose the recipe tokens ($t_R$) as queries, as the number of recipe tokens ($N_R \sim 45$) is much less than the number of image tokens ($N_I = 196$), which reduces the complexity to $\sim N_R^2 + N_R \times N_I$ in contrast to $\sim (N_R + N_I)^2$ for a transformer encoder.

The MMR module consists of inter-connected transformers; an Image Tokens Enhancement Module (ITEM) to enhance the image tokens, followed by a multimodal transformer decoder (MTD):
Image Tokens Enhancement Module (ITEM) We propose to enhance the image tokens before entering MTD, by adding a transformer decoder that takes the image tokens as queries (Q) and the recipe tokens as Keys and Values (KV):

\[ \hat{t}_I = ITEM(t_R, t_I). \]  
(2)

This module enriches the image tokens by attending to the recipe elements. While it seems natural to use only the ingredients, as they are present in different regions/tokens of the image, we found that other recipe elements, even if they are not explicitly presented in the image, brings additional improvements (see supplementary material).

Multi Modal Transformer Decoder (MTD) Unimodal encoders encode each modality independently, and lack the ability to capture the interaction between the two modalities. Here we propose a multimodal transformer decoder (MTD) that predicts the matching score of each pair of samples. We argue that this predicted matching score better captures the interaction between the two modalities, compared to the global cosine similarity applied between the independently extracted embeddings. Specifically, MTD contains self-attention, cross-attention and feedback modules. For the cross attention, the tokens of one modality are considered as queries (Q) and the tokens of the other modality as Keys and Values (KV). The matching score between a pair of samples can be obtained as follows:

\[ s(t_R, t_I) = MTD(t_R, \hat{t}_I) = MTD(t_R, ITEM(t_I, t_R)), \]  
(3)

where \( t_R \) and \( \hat{t}_I \) are the output recipe tokens and the enhanced image tokens respectively.

Image-Text Matching (ITM) loss: ITM is a binary cross-entropy loss that is optimised to classify whether or not an image-text pair matches. Following the success of batch hard mining for triplet loss, the loss is applied on the hardest negative sample; for each recipe/image anchor we sample a negative recipe/image in the batch that is most similar to the anchor using the fast cosine similarity of the embeddings at the output of the DE. The loss can be written as follows:

\[
\mathcal{L}_{itm} = -\mathbb{E}_{t_R, t_I \sim D}[y \log(s(t_R, t_I)) + (1 - y) \log(1 - s(t_R, t_I))],
\]  
(4)

where \( y \) is the label (i.e., 1 for matching pair and 0 otherwise) and \( D \) is the set of pairs.

3.3. Retrieval loss

Following other work on cross-modal food retrieval [20, 47, 48, 58], we use the triplet loss in our learning framework. Here we introduce a new variant of triplet loss with dynamic margin.

**Triplet loss:** The triplet loss can be written as:

\[
l(x_a, x_p, x_n, \alpha) = [d(x_a, x_p) + \alpha - d(x_a, x_n)]_+, \]  
(5)

where \( x_a, x_p \) and \( x_n \) are the anchor, positive and negative examples respectively, \( \alpha \) is the margin and \( d(\cdot, \cdot) \) is a distance function.

**IncMargin loss:** Inspired by the curriculum learning approaches [3], where the task becomes harder as the training progresses, we propose to replace the constant margin by a dynamic one. The idea is that the difficulty of the task is also affected by the margin; a small margin is easier to optimize than a large margin. Here we assume that the task is difficult at the beginning, thus we make it simpler by starting with a small margin \( \alpha_{inc} \) and increase it at each epoch until reaching a maximum value:

\[
\mathcal{L}_m(B_a, B_p, B_n) = \sum_{x_a \in B_a} \sum_{x_p \in B_p} \sum_{x_n \in B_n} l(x_a, x_p, x_n, \alpha_{inc})
\]  
(6)

We keep the adaptive margin between an acceptable minimum and maximum values. In addition, we follow Adamine [6] and use the triplet loss with an adaptive weighting strategy, where the triplet is weighted by a similar term as \( \delta \) to overcome the vanishing update when most of the triplets are inactive. We call the instance loss as image-text contrastive loss which can be written as:

\[
\mathcal{L}_{itc} = \frac{1}{\delta_r} \mathcal{L}_m(B^r_a, B^r_p, B^r_n) + \frac{1}{\delta_v} \mathcal{L}_m(B^v_a, B^v_p, B^v_n)
\]  
(7)

Where \( B^r \) and \( B^v \) are the sets of recipe and image embeddings, \( \delta_r \) and \( \delta_v \) are the number of triplets that contribute to the loss \( (l > 0) \text{ Eq. } 5 \) when the recipe and image embeddings are anchors respectively. The positive and negative samples are from the other modality; the positive is the one associated with the anchor and other samples in the batch are considered negatives.

Similarly, we incorporate the semantic triplet loss of Adamine [6] as a regularization to obtain more semantically rich embeddings. The semantic loss \( \mathcal{L}_{sem} \) is the same as the instance loss except for the selection of positive and negative samples. Here the positive samples are those that share the same class with the anchor and the negative samples are the samples with different class.

**Total loss:** Thus, the total loss can be written as follows:

\[
\mathcal{L} = \mathcal{L}_{itc} + \lambda_{sem} \mathcal{L}_{sem} + \lambda_{itm} \mathcal{L}_{itm}
\]  
(8)

Where \( \lambda_{sem} \) and \( \lambda_{itm} \) are the corresponding losses weights.
4. Experiments

We use Recipe1M dataset [48] with 238,999, 51,119, 51,303 pairs as training, validation and test set. We report median rank (medR) the median of the retrieved samples index for each query, and recall@K (i.e., R@1, R@5, R@10) the percentage of queries for which the correct sample index belongs to the top K retrieved samples. We report the mean over 10 (resp.5) bags of 1k (resp. 10k) pairs of the Recipe1M test set.

Model details: We use ViT-B/16 and CLIP-ViT-B/16 as image encoders. For the recipe encoder, similarly to [47], we use transformer encoders with 2 layers and 4 heads for hierarchical transformers $T$ and $HT$. For $HTD$ we use a transformer decoder (without masking) with 2 layers and 4 heads. The hidden layer dimension is kept 512 in the recipe encoder. The image and recipe embeddings are obtained with a different linear layers of output dimension 1024. The image and recipe tokens are projected using different linear layers of the same output dimension 1024 before going to the $MTR$ module. To keep the model simpler, the $ITEM$ module consists of a transformer decoder of only 1 layer and 4 heads with hidden size 1024. The $MTD$ consists of a transformer decoder with 4 layers, 4 heads and hidden dimension 1024. All the transformers are trained from scratch, except the ViT, which is initialised by ImageNet weights, and CLIP-ViT with CLIP weights.

Training details: We mainly follow the implementation details of Adamine [6]. Specifically, For the alignment loss we use $\lambda_{sem} = 0.1$ for the semantic triplet loss and $\lambda_{itm} = 1$. For each batch, half of the samples are associated with a class for the semantic loss. To compute the triplet loss, each sample is considered as anchor, the associated sample from the other modality as positive sample and all the others samples from the other modality in the batch as negatives. For the semantic triplet loss, the samples with the same class are considered positive samples and negative samples otherwise. The image encoder is kept frozen for the first 20 epochs, then all the modules are trained with a constant learning rate $1e^{-5}$ (except CLIP-ViT with learning rate $1e^{-6}$) and Adam optimiser. The models are trained until reaching 120 epochs with batch size 100. The models are trained with 2 A100 GPUs 40 GB. For IncMargin, we start by a $\alpha_{inc} = 0.05$ and increase it by 0.005 each epoch until reaching 0.3. We use a margin $\alpha = 0.3$ for all other experiments.

4.1. Comparison with Other Work

We compare our work to existing approaches in Table 2. For Adamine, we shifted the values of medR as they assume medR=0 is for exact match, while other approaches adopt medR=1, we also notice that their metrics for recipe to image and image to recipe should be swapped. For HT [47], we use their official code to re-compute the metrics with the ViT image encoder (H-T (ViT)) and we report these metrics for fair comparison with our approach. We can notice that our approach significantly outperforms all existing work. Best metrics are in bold, and next best metrics are underlined.
Table 3. Ablation study. medR (↓), Recall@k (↑) are reported on the Recipe1M test set with 1k setup. Each modification brings additional improvement as well as the combination of all of them.

<table>
<thead>
<tr>
<th></th>
<th>image-to-recipe</th>
<th>recipe-to-image</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>medR R@1 R@5 R@10</td>
<td>medR R@1 R@5 R@10</td>
</tr>
<tr>
<td>✔</td>
<td>59.5 83.1 88.1</td>
<td>61.1 83.5 88.2</td>
</tr>
<tr>
<td>✔ ✔</td>
<td>65.6 86.7 90.8</td>
<td>66.8 86.9 90.8</td>
</tr>
<tr>
<td>✔ ✔ ✔</td>
<td>66.9 87.0 91.0</td>
<td>67.5 87.2 91.1</td>
</tr>
<tr>
<td>✔ ✔ ✔ ✔</td>
<td>67.0 87.5 90.9</td>
<td>68.1 87.4 91.0</td>
</tr>
<tr>
<td>✔ ✔ ✔ ✔</td>
<td>68.2 87.9 91.3</td>
<td>68.3 87.8 91.5</td>
</tr>
<tr>
<td>✔ ✔ ✔ ✔ ✔</td>
<td>72.3 90.7 93.4</td>
<td>72.6 90.6 93.4</td>
</tr>
</tbody>
</table>

Table 4. Ablation study for AdaMargin and IncMargin on the Recipe1M test set with 1k setup. We compare our baseline model B (Sec. 4.2) to three variants trained without Adamine. Our new triplet loss IncMargin outperforms significantly AdaMargin and the classical triplet loss with fixed margin. Both AdaMargin and IncMargin outperforms B.

<table>
<thead>
<tr>
<th>Margin scheme</th>
<th>image-to-recipe</th>
<th>recipe-to-image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline B</td>
<td>medR R@1 R@5 R@10</td>
<td>medR R@1 R@5 R@10</td>
</tr>
<tr>
<td>Fixed α</td>
<td>59.5 83.4 88.1</td>
<td>60.7 83.5 88.3</td>
</tr>
<tr>
<td>AdaMargin α/δ</td>
<td>60.8 83.4 88.2</td>
<td>61.5 83.7 88.4</td>
</tr>
<tr>
<td>IncMargin α_{inc}</td>
<td>62.9 84.8 89.4</td>
<td>63.8 85.1 89.3</td>
</tr>
</tbody>
</table>

4.2. Ablation Study

In this section, we investigate the importance of different design choices. Our baseline (B) consists of the dual encoders: ViT as image encoder, hierarchical transformer as recipe encoder (i.e., with T and HT), and Adamine loss.

From Table 3 we can notice that the proposed HTD leads to better alignment, seen by the significant improvement of metrics, this validates our hypothesis that the recipe elements are highly entangled and should not be embedded separately. We can also notice that MTD also brings additional improvement, this is consistent with the findings of VLP in general and reveals that multimodal transformers are promising for the underlying task.

ITEM brings additional improvements, which in some sense facilitates the role of MTD, by fusing further the image with the recipe tokens. Using ViT-CLIP as image encoder brings significant improvement, this indicates that using models trained on large datasets can cope with the noise in this relatively small dataset.

From table 4 and 3, our IncMargin loss, based on an increasing margin value α_{inc}, improves upon the classical triplet loss by more than 3 points (image to recipe R@1). In addition, we compare to AdaMargin, another variant of triplet loss with adaptive margin that changes according to the number of active triplet in the batch (see supplementary material). Despite its simplicity, IncMargin also outperforms AdaMargin. From our experiments, we noticed that
the value of the margin increases to its maximum value faster with AdaMargin, while the evolution of the margin is smoother and takes longer time to reach its maximum value with IncMargin. In addition, we can notice that the improvement of this variant are smaller with the ITM loss (from 67.0 to 68.2 R@1 image-to-recipe), this should be expected as the model focuses on the ITM regularization when the triplet loss is small (i.e., the margin is small at the beginning of the training) and we noticed from our experiments that the triplet/retrieval loss is more important than the ITM for retrieval.

\[
\text{R@1 } \quad \text{R@5 } \quad \text{R@10}
\]

65.6 86.7 90.8

67.8 87.6 91.3

66.8 87.0 90.4

67.5 87.5 91.0

67.0 86.9 90.4

66.1 86.5 90.3

Table 5. Ablation study for MMR on the test set with 1k setup.

We report the performance w.r.t the number of layers in MTD. All the models are trained with \( B + HTD + MMR \) and \( \lambda_{itm} = 0.1 \).

MMR: To perform image/recipe retrieval, we compute the pairwise cosine similarity of each pair of image/recipe in the shared embedding space, which can be done efficiently in parallel. Other work [30, 41] propose to use multimodal modules to re-rank the top \( k \) retrieval propositions for each query. Since these multimodal modules compute a more fine-grained image/text alignment, it improves the retrieval metrics, but with a much higher computational cost. In Figure 4, we compare those two retrieval methods for our model variants that use a MMR module. We can see that our model MTD w/o Adamine has a low R@1 score without re-ranking (54.0) but much higher with multimodal re-ranking (63.6), emphasizing the role of multimodal alignment in that case. Adamine helps to partially close this gap by giving more importance to the shared embedding space, which improves greatly the retrieval metrics without re-ranking, with small improvements of the metrics with re-ranking.

When we add the HTD module (for recipe encoding), we observe that the role of re-ranking becomes useless. This remains true for our best models T-Food and T-Food (CLIP). Therefore, we conclude that the improvement brought about by re-ranking was due to the lack of expressiveness of the recipe encoder and not the need for a finer image/text alignment. We also conclude that the presumably better alignment capabilities of the MMR module are fully transferred to the shared embedding space at the end of training, given that re-ranking with this module is not better than global embedding similarity. We also investigated how the complexity of MMR affects the performance in Table 5, where we train our model with different number of layers in MTD.

We can notice that when MTD becomes more powerful, the metrics become better (nb. of layers = 1 vs 4). Note that, these results are consistent with other contrastive learning work, specifically, in SimCLR [8], the authors show better results when the projection head is more powerful (e.g., nonlinear instead of linear), and here we make the analogy between their projection head and our MRR module. However, a too heavy MTD degrades the results (nb. of layers = 4 vs 6), we argue that when the regularization module becomes too powerful, most of the modality fusion and cross modal interaction will be focused in it, instead of the DE.

We can notice that each modification brings additional improvement as well as the combination of them.

Acknowledgments: This work was partly supported by ANR grant VISA DEEP (ANR-20-CHIA-0022), and HPC resources of IDRIS under the allocation 2021-AD011013159 made by GENCI.

5. Conclusion

We proposed a new framework for multimodal alignment in the cooking context and validated the approach by surpassing all existing work by a large margin on the cross modal retrieval task. We introduced a novel MultiModal Regularization and new variant of triplet loss, in addition we showed that transformer encoders/decoders can bring significant improvement by exploiting the intra and inter dependencies between modalities and recipe elements. Finally, we showed that transfer learning from VLP models trained on large datasets can significantly help in this context. While the work is successful in the food domain, we believe that the proposed ideas can be beneficial for cross modal tasks in general, especially cross modal retrieval.
References


[26] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In International Conference on Machine Learning, pages 4904–4916. PMLR, 2021. 2


[34] Lin Li, Ming Li, Zichen Zan, Qing Xie, and Jianquan Liu. Multi-Subspace Implicit Alignment for Cross-Modal Retrieval on Cooking Recipes and Food Images, page 3211–3215. Association for Computing Machinery, New York, NY, USA, 2021. 3, 6, 7


[71] Xiaonan Zhao, Huan Qi, Rui Luo, and Larry Davis. A weakly supervised adaptive triplet loss for deep metric learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops, pages 0–0, 2019. 3