Food Images to Recipes using Cross-Modal Retrieval

Anmol SHARMA sharma265@wisc.edu
Pengyu ZHANG pzhang256@wisc.edu
Xinmiao XIONG xxiong52@wisc.edu

Final Report of BMI/CS771 Course Project
All team members contributed equally

Abstract – In this project,we are using Cross-Modal Retrieval method which would give out the possible recipes any dish would have when provided with the image of the dish. We have upgraded the loss function based on the nature of the Recipe1M dataset and done a comparable study of different image encoders based on the muti-LSTM recipe encoder method. The code is available here: https://github.com/2ndBlossom/image2recipes

1 Introduction

Food is paramount to humans. It has deep connections to our health, our emotions and our cultures. Thus many recognition studies of food images have been done in computer vision field, including but not limited to automatically determining the calories in a dish, commenting dishes in ranking systems, ingredients prediction and recipe recommendation. In this project, we investigated the issue of cross-modal retrieval between food images and text-based cooking instructions.

After the release of the Recipe1M dataset[1], cross-modal retrieval became one of the baseline for image-to-text tasks for food recipe retrieval. Recipe1M dataset contains over 1M cooking recipes and 800K food images. It is easy to find that the quantities of images and recipes don't match. There are many recipes don't have the paired images. Thus we can divide the dataset into two parts, images with paired recipes, and non-image recipes. The later part only contains text message.

Based on the baseline model, we committed a comparable study of different image encoders' performance in this cross-modal retrieval problem. Considering there are two types of data in our dataset, we implemented a combined loss function in our model. The combined loss contains a supervised loss(recipes with images) and a self-supervised loss(recipe without images). Due to the limitation of computation, we used a subset for experiment.

We have equal contribution for this project, and here is our github link for the project code:

2 Related Work

Cross-modal retrieval aims to pair relevant instances from different modality, such as retrieving an image using text. Encoding images and text using pre-trained convolutional image recognition models and Long short-term memory (LSTM)[2] or Transformer[3] text encoders is typically used in methods designed for this task.

Encoding cooking recipes, which are long and structured texts, can be challenging. Some previous studies have suggested encoding each recipe component separately and then combining them using late fusion techniques to create a fixed-length recipe representation.

There is currently no agreed-upon approach for utilizing recipe information or determining the best encoders for each component. Many previous studies have neglected to include the recipe title in their encoding efforts[4,5]. Many previous studies have chosen to use Long short-term memory (LSTM) encoders for their architecture, using single LSTMs to encode individual sentences (e.g., titles, ingredient lists) and hierarchical LSTMs to encode sequences of sentences (e.g., raw ingredient lists or cooking instructions).

It should be noted that while the Recipe1M dataset is multimodal, only 33% of the recipes have accompanying images. Previous research has typically only utilized the paired samples to optimize the image-recipe space, ignoring the text-only samples or using them only for pre-training text embeddings[3,4,5]. To improve the performance of our model under certain metrics, we implemented a self-supervised loss based on the recipe representations generated by our model. This allowed us to train with additional recipes that were not linked to any images.

3 Data

As the food images are very complex and understanding of the features and textures of the food present in an image is very difficult. The Recipe1M dataset has been divided into two dataframes. The first dataframe contained information such as the id, ingredients, title, instructions, and original URL for each of the Recipe1M+ set's 1 million recipes. Along with a list of the image ids connected to each recipe, the second dataframe also included the ids of the recipes in the Recipe1M set. The second dataframe was changed by extracting all of the individual image ids and replacing them with new dataframes that con-

nected each image id to the corresponding recipe. The images were also preprocessed by being resized to a shape of (256, 256, 3) before being used in any CNN models. From these one million pool of data, a small subset of 20000 were used in this project. And out of which we have only 60 percent of paired data(image-recipe) and rest 40 percent non-paired recipes.

4 Training

The dataset in divided into training, testing and validation in the ratio 70:15:15. Futhermore, for training a batch size of 40 were chosen and loss was minimized using ADAM optimizer. Learning rate is selected based on performance of model after 10 epochs out of 100 epochs. The selected learning rate is 0.0001 for ResNet-50, 0.001 for DenseNet-121, 0.001 for ViT. The shortest side of the images was resized to 256 pixels and then the images were cropped to 224 x 224 pixels. For recipe-only items, we increased the batch size to 70 as less GPU memory is required when we exclude the image encoder."

5 Approach

In this project we trained a joint neural embeddings on the dataset. We evaluated by comparing the performance of various image and recipe encoding models. Our method's complete network architecture is depicted in Figure 1.

5.1 Image Embeddings

The work of the image encoder is to project the input image into the joint image-recipe embedding space. Here, in this project, ResNet-50, VGG16 and DenseNet101 are used to embed the images and then, transfer learning methods are implemented to accelerate the training process by using CNNs that were pre trained on ImageNet. These models are incorporated by subtracting the last softmax classification layer and connecting the rest to the joint embedding model. For the improvement, we also use Vison Transformer for the image encoder part. Here we used the ViT-B/16 pretrained model. Then we can gain the representations e_I for the images after image encoding.

5.2 Recipe Embeddings

The purpose of the recipe encoder is to project the input textual recipe data into the joint image-recipe embedding space. Here we use LSTM-based encoder. There are three parts of each recipe data: tittle, ingredients, instructions. And the nature of each part is different, so we have to develop suitable representations for each of these three components.

For tittle, it contains a single sentence, so it is easy to build it's representation e_{ttl}. To build ingredient-level word2vec representations for each recipe, we used a bi-directional Long Short-Term Memory (LSTM) model to perform logistic regression on the words in the ingredient text, as shown in Figure 1. Here we get the representations for ingredient e_{ing}

For cooking instructions, it is even more complicated. Here we used a two-stage hierarchical LSTM encoder, which can encode inputs made up of sequences of sentences. The first stage LSTM model takes a list of M sentences as input and generates M fixed-sized embeddings, one for each sentence in the list. The second stage LSTM model accepts as input a list of sentence-level embeddings and outputs a single representation $eins$ for the full list of sentences. Figure 2 depicts the structure of this hierarchical LSTM model.

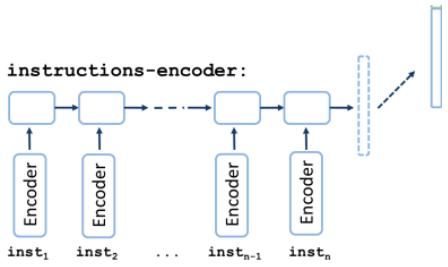

Figure 2: Hierachical LSTM for Instructions

5.3 Joint Embeddings

The image encoding and the recipe encoding that is performed in the previous step are then mapped to a joint space of embedding to form the overall recipe encoding which is then used to obtain the performance evaluation of the model. From the image set, with a random probability of 25percent the matching image-recipe pair is selected and with the remaining 75 percent of probability the non-matched up pair is chosen. For the loss function, Cosine similarity loss[1][2] was used where the model is trained end-to-end with the negative and positive image-recipe pairs. For loss function, we used a combined loss which contains two parts, supervised loss L_{pair} for paired data and self-supervised recipe loss L_{rec} for unpaired recipe data.

5.4 Loss Function

The intuition of our use of combined loss function is based on the characteristics of the dataset Recipe1M. Firstly, it is the presence of unpaired text-only data. In Recipe1M, 40% of the whole dataset consists of items that don't include images, which means these data only contain a textual recipe. We don't want to ignore the large amount of unpaired data when we are training our joint embeddings. Second, while the separate components of a recipe (such as the title, ingredients, and instructions) contribute complementary information, they also share significant semantic cues that can be leveraged to build more robust and semantically consistent recipe embeddings. In order to do so, we built a self-supervised loss L_{rec} for every recipe which constrains recipe embeddings so that the intermediate representations of a dish's constituent components should be close together if they are from the same recipe, and far apart if they are from distinct recipes.

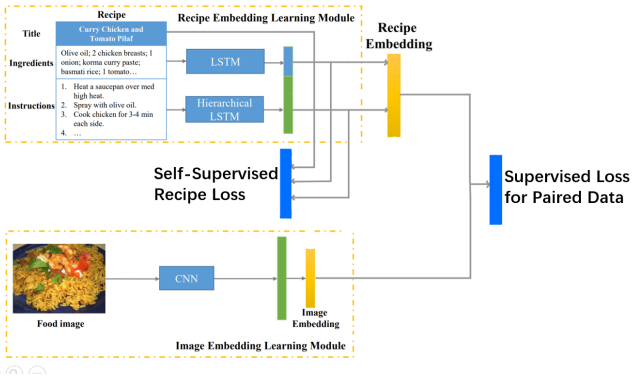

Figure 1: Overview of Cross-modal Retrieval Method

5.4.1 Supervised Loss L_{pair} for paired data

We have used triplet loss because of its great success and its representation is as follows:

$$L_{\cos}(an, p+, n-) = \max(0, c(an, n-) - c(an, p+) + margin) \tag{1}$$

Where an, $p+$, and $n-$ denote the anchor, positive, and negative samples, $c(*)$ is the cosine similarity measure, and the margin for all triplet losses utilized in this project is empirically set at 0.3. On feature sets x and y, we use the bi-directional triplet loss function Lbi' in practice.

$$\begin{aligned} L'_{yi}(i,j) = L_{\cos}\left(x^{n=i}, y^{n=i}, y^{n=j}\right) + \\ L_{\cos}\left(y^{n=i}, x^{n=i}, y^{n=j}\right) \end{aligned} \tag{2}$$

Here $x^{n=i}$ is positive to $y^{n=i}$ and negative to $y^{n=j}$. During training, the loss for each sample i in a batch of size A is the average of all losses, with all other samples in the batch treated as negatives, as shown below.

$$L_{bi}\left(x^{n=i}, y^{n=i}, y^{n\neq i}, x^{n\neq i}\right) = \frac{1}{A}\sum_{j=0}^{A} L'_{yi}(i,j)\delta(i,j) \tag{3}$$

Here $\delta = 1$ if $i \neq j$ and $\delta = 0$ if $i = j$.We calculate the paried loss L_{pair} by using L_{bi}. Here e_I and e_R are representations extracted by image and recipe encoders.

$$L_{\text{pair}} = L_{bi}\left(e_I^{n=i}, e_R^{n=i}, e_R^{n\neq i}, e_I^{n\neq i}\right) \tag{4}$$

5.4.2 Self-supervised Recipe Loss, L_{rec}

Here e_{ttl}, e_{ing}, e_{ins} are representations of title, ingredients, instructions extracted by recipe encoders. As shown in the Figure 2. For example, we first use a single linear layer $g_{ttl\to ing}$to project e_{ttl} to e_{ing} feature space as $e_{ttl\to ing}$.

$$e_{ttl\to ing} = g_{ttl\to ing}\left(e_{ttl}\right) \tag{5}$$

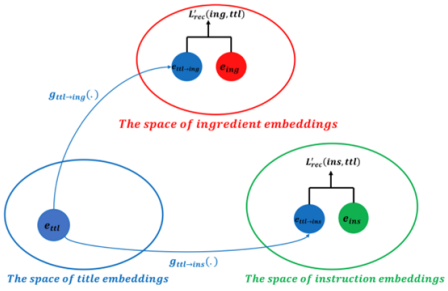

Figure 3: Projection Function

Then we can calculate the the L'_{rec} by using the L_{bi} we used before. And thus we make e_{ttl} closer to its corresponding ingredient representations and farther from the ingredients from different recipes.

$$L'_{\text{rec}}\left(ing, ttl\right) = L_{bi}\left(e_{ttl}^{n=i}, e_{ttl\to ing}^{n=i}, e_{ttl\to ing}^{n\neq i}, e_{ttl}^{n\neq i}\right) \tag{6}$$

As shown in the Figure 3, there are six variation of projection functions for all combinations of title, ingredients and instructions. Finally, we compute all possible combinations and average the result and get the final self-supervised recipe loss funstion L_{rec}.

$$\begin{aligned} L_{rec} = \frac{1}{6.0}\sum_{x}\sum_{y} L'_{rec}(x,y)\delta(x,y), \\ x, y \in \{ttl, ing, ins\} \end{aligned} \tag{7}$$

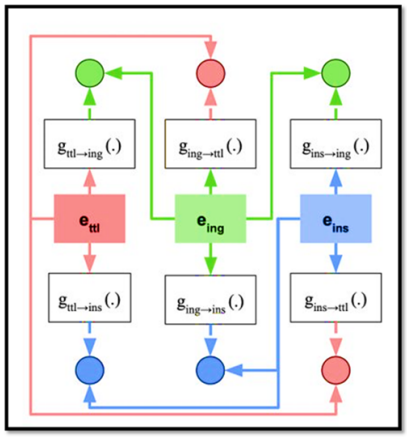

Figure 4: Self-supervised Recipe Losses

The combined paired loss and recipe loss are what make up the final loss function, which is denoted by the formula $L = alphaLpair + betaLrec$. For paired samples, both $alpha$ and $beta$ are assigned to 1.0, while for text-only samples, $alpha = 0.0 and$beta = 1.0.We trained our model using self-supervised triplet loss on recipe components and tested its performance in scenarios where some data was missing. We found that using projection layers, which allow the model to "hallucinate" missing recipe components by taking the average of the projected vectors of the remaining components, resulted in improved performance when some recipe data was missing at test time.

6 Results

Modals	MedR	R@1	R@5	R@10
ResNet50	12.0	0.16	0.33	0.42
DenseNet-121	11.1	0.20	0.36	0.47
ViT	10.5	0.21	0.37	0.49

Table 1: Comparison of Image Encoders: Performance reported on validation dataset, and on rankings of size 1k

Loss Function	MedR	R@1	R@5	R@10
Cosine Loss	15.4	0.109	0.31	0.44
Combined Loss	11.1	0.280	0.369	0.49

Table 2: Desnet-121 performance Comparison with different loss function

We evaluated retrieval performance using the median rank (medR) and Recall@1, 5, 10 (also known as R1, R5, and R10), as has been done in previous research." Here, medR is the median rank of true positives returned. As a result, high performance is associated with a lower medR score. R@k estimates the proportion of times the proper recipe is discovered among the top-k retrieved recipes given a food image. It's value is directly proportional to modal performance. In image encoding Densenet-121 performed better than Resnet-50 because of better information and gradient flow, also it is easy to manage very deep layers in DenseNet. Overall, ViT outperforms the other model because of the self-attention layers which help the network to learn the hierarchies and alignments present inside the image. In Table 2, we compare the performance of our new loss to that of the previous loss by using DenseNet-121 model on the validation set and evaluating its performance on the test set.. When trained with only paired data, our model's combination loss modal outperforms one with cosine loss. When we include the additional unpaired data without photos, the retrieval accuracy improves even more (R1 of 0.28 and R5 of 0.36), which uses triplet loss to embed pre-learned recipe embeddings (training is done on complete training set) and pre-trained picture representations.

7 Discussion

Although good results were achieved with the ViT image encoder and LSTM text encoder ,but can't comparable with performance with benchmark, so we plan to compare the performance based on our subset using different models. This is because the dataset used in this project is only around 1/20 of the size of the original dataset. Also, the variance of food images due to the reduced dataset hampers the performance of the model. So it was concluded that to do such a task there is a requirement for a huge training dataset. Another shortcoming of the model is the inability to train large batch sizes, which is known to lead to fewer parameter updates and greater parallelism. In this project a batch size of 50 was used which is almost half used in the original paper due to GPU constraints. It would be interesting to investigate whether the current algorithm, which was developed for food recipes, could be applied to other types of recipes that involve assembling guidance/instructions and materials, given an image of the finished product. Also,to make it a more complete work, we can add GAN-based architecture to enhance the cross-modal embedding learning in addition to adopting Transformer-based recipe encoders.

References

[1] Salvador, Amaia, et al. "Learning cross-modal embeddings for cooking recipes and food images." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[2] Graves, Alex. "Long short-term memory." Supervised sequence labelling with recurrent neural networks (2012): 37-45.

[3] Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).

[4] Salvador, Amaia, Hynes, Nicholas, Aytar, Yusuf, Marin, Javier, Ofli, Ferda, Weber,Ingmar, and Toralba, Antonio. Learning cross-model embbeddings for cooking recipes and food images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[5] Wang, Xin, et al. "Recipe recognition with large multi-

modal food dataset." 2015 IEEE International Conference on Multimedia Expo Workshops (ICMEW). IEEE, 2015.

[1] Salvador, Amaia, Hynes, Nicholas, Aytar, Yusuf, Marin, Javier, Ofli, Ferda, Weber,Ingmar, and Toralba, Antonio. Learning cross-model embbeddings for cooking recipes and food images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[2] Chen, Jianfa, Yue Yin, and Yifan Xu. "RecipeSnap–a lightweight image-to-recipe model." arXiv preprint arXiv:2205.02141 (2022).

[3] Fain, Mikhail, et al. "Dividing and conquering cross-modal recipe retrieval: from nearest neighbours baselines to sota." arXiv preprint arXiv:1911.12763 (2019).

[4] Salvador, Amaia, et al. "Revamping cross-modal recipe retrieval with hierarchical transformers and self-supervised learning." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2021.

[5] Marin J, Biswas A, Ofli F, et al. Recipe1m+: A dataset for learning cross-modal embeddings for cooking recipes and food images[J]. IEEE transactions on pattern analysis and machine intelligence, 2019, 43(1): 187-203.