Text and Vision Transformers for Image-to-Recipe Retrieval

Malthe Dohm Andersen and Yucheng Fu

Technical University of Denmark, DTU, November 2023 Project, Advanced Deep Learning in Computer Vision

Data

We are working with the "Food Ingredients and Recipes Dataset with Images" from Kaggle [1]. The dataset contains 13501 images, and a CSV file containing Recipe Title, Ingredients, Instructions, Image name and "Cleaned Ingredients". The CSV has 13501 rows, with 30 rows missing an image name. The rows with missing image names were removed from the dataset during preprocessing.

A few image samples are visualised below:

(a) (b) (c) (d)

Figure 1: Example of images from the Food Ingredients and Recipes Dataset.

While the majority of images are of food dishes [4a,4b] as we would expect, some images are from the cover a cookbook [4c] or seemingly unrelated [4d].

The data is partitioned into a training, validation, and testing set containing 10776, 1617 and 1078 samples respectively.

Model Architecture

The overarching idea for the model architecture is based on the single idea of projecting text embeddings and image embeddings to a common space, so that the distance from an image to a text can be measured. While the image encoder has a simple solution, the text is made up of separate texts (title, ingredients, instructions).

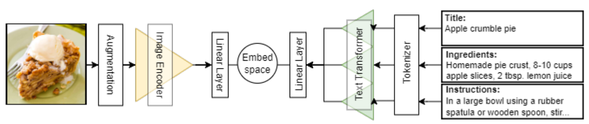

Figure 2: The "late-fusion" model architecture. The individual text parts are tokenized individually, then passed to the text transformer. The embeddings are concatenated before being passed to the linear layer, which acts as a projection head that projects text and image into common embedding space.

The model uses "late-fusion", which references the fact that the different text modalities are concatenated after being encoded.

Encoders

Text encoders:

- BERT - pretrained
- Text transformer - non-pretrained

Image encoders:

- ResNet-50 - pretrained
- vision transformer - non-pretrained

Triplet Loss

Triplet Loss is a loss function that seeks to minimize the distance from an anchor A, e.g. the input image, to a positive pairing P, in this case the correct title, while maximizing the distance to negative pairings N, the other incorrect titles. For each anchor-positive pair the loss is calculated as

$$\mathcal{L}(A, P, N) = \max(d(A, P) - d(A, N) + m, 0)$$

where d is the cosine distance and m is a margin that keeps the negative and positive samples apart.

Training

An ablation study is performed to test different model choices and investigate how they impact the performance of the model. The model choices in the ablation study are shown below:

Choice	Values
Text modality	{Title, Title + ingr., Title + ingr. + instr. }
Data augmentation	{ True, False}
Embed dim	{64, 128, 256}
Learning rate	{1e-5, 1e-4, 3e-4}
Margin	{0.1, 0.3, 0.5}

Table 1: Model choices in ablation study. Underline denotes default parameters.

The data augmentations are inspired by [3]. During training:

- Images are resized to 256×256
- Cropped at random location to 224×224
- Randomly flipped horizontally with $p = 0.5$

In total, 11 models were trained. 10 models are trained using pretrained encoders with frozen weights, while the last one is our architecture trained from scratch.

During training, the loss is computed for both image features as anchors, denoted $\mathcal{L}_{\text{image}}$ and text features as anchors, denoted $\mathcal{L}_{\text{text}}$. The total loss is the sum of both, averaged for all pairs in the batch:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{image}} + \mathcal{L}_{\text{text}}$$

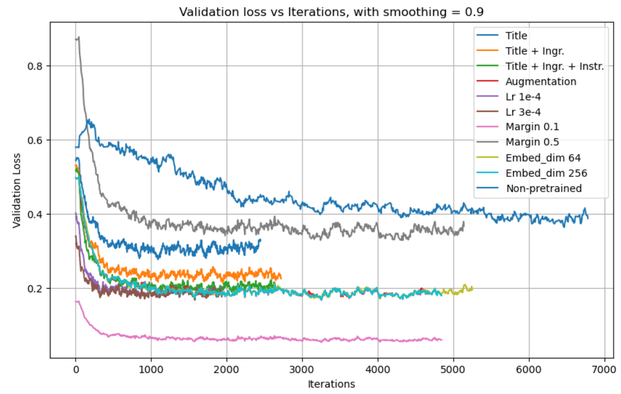

Figure 3: Loss graphs of different models for validation set

Evaluation

During inference, the images are only cropped to 224×224 at the centre.

For quantitative evaluation, the median rank is calculated, along with the recall for the top 1, top 5 and top 10 guesses.

We use wordclouds to visualize text next to its most similar images and vice versa.

References

[1] https://www.kaggle.com/datasets/pes12017000148/food-ingredients-and-recipe-dataset-with-images.

[2] https://github.com/openai/CLIP.

[3] Amaia Salvador, Erhan Gundogdu, Loris Bazzani, and Michael Donoser. Revamping cross-modal recipe retrieval with hierarchical transformers and self-supervised learning, 2021.

Quantitative results

For the models in the ablation study, the median rank (medR) and the top k accuracy (R @ k) measures are computed. Each model is evaluated on both the image-to-text and text-to-image performance. A pretrained CLIP[2] model is used to produce text and image embeddings. The text embeddings from CLIP are produced using only the title.

Image-to-text

	medR	R @ 1	R @ 5	R @ 10
Non-pretrained	350.0	0.093	0.84	1.94
Title	147.0	0.93	4.91	8.16
Title + ingr.	98.0	1.58	6.86	12.43
Title + ingr. + instr.	82.5	2.41	9.28	14.84
Augment	**63.5**	2.97	**12.89**	**18.83**
Emb 64	**63.5**	2.97	**12.89**	**18.83**
Emb 256	**63.5**	2.97	**12.89**	**18.83**
Lr 1e-4	69.0	2.60	10.48	18.46
Lr 3e-4	68.0	**3.80**	11.60	17.81
Mar 0.1	64.5	**3.80**	11.87	18.46
Mar 0.5	69.5	2.78	11.13	16.79
CLIP	3.0	30.89	59.28	68.74

Table 2: Image-to-text metrics on entire test set.

Text-to-image

	medR	R @ 1	R @ 5	R @ 10
Non-pretrained	355.0	0.093	1.21	2.04
Title	152.5	1.58	5.19	8.63
Title + ingr.	102.0	2.04	7.14	12.43
Title + ingr. + instr.	83.5	2.32	8.81	14.94
Augment	**63.0**	3.90	**13.36**	**20.78**
Emb 64	**63.0**	3.90	**13.36**	**20.78**
Emb 256	**63.0**	3.90	**13.36**	**20.78**
Lr 1e-4	72.5	3.34	11.78	19.76
Lr 3e-4	71.0	3.06	11.60	17.81
Mar 0.1	70.0	**4.08**	12.99	19.67
Mar 0.5	70.5	3.34	12.06	18.46
CLIP	3.0	32.47	61.13	71.80

Table 3: Text-to-image metrics on entire test set.

Qualitative Results

In order to see what text the best model predicts for some example images and recipes, we find the closest text and image in the embedding space.

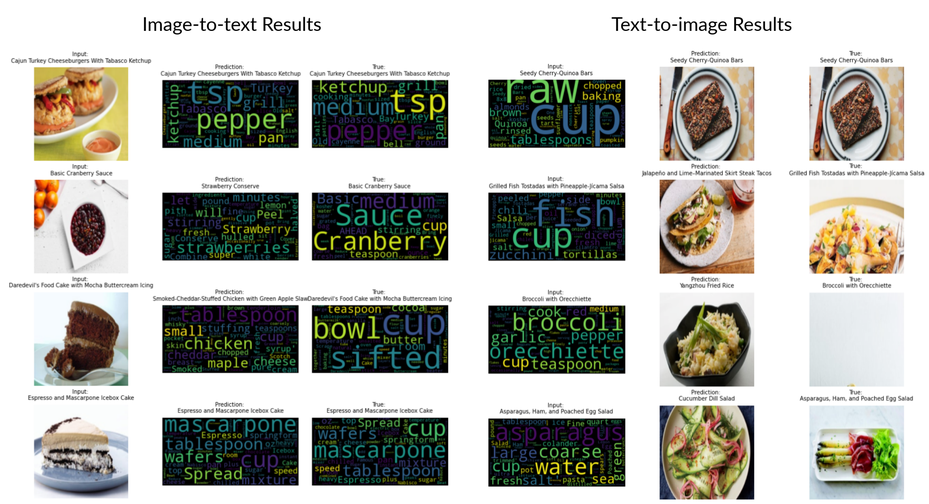

Figure 4: Example of images and recipes from the Food Ingredients and Recipes Dataset, retrieved by the model.

Discussion

Incorporating more text modalities improves the performance of the image-to-recipe retrieval and recipe-to-image retrieval tasks. Data augmentation also greatly improved the performance. Among the model choices, we found that the model with margin 0.1 resulted in the best R @ 1.

Poor performance could be due to the quality of the dataset. We noticed that there were several plain white images and images that were cropped weirdly such that only a small corner of the dish was visible.

Training from scratch did not yield good results, perhaps because learning to embedding features directly into the shared space is too difficult of a task.

For further research one could try finetuning the pretrained encoders while training the projection heads. However this will require more lightweight pretrained encoders, since the combination of BERT and ResNet-50 proved too costly in terms of memory.