Cross Modal Representation Learning Final Report

Kunj Mehta
Rutgers University
Department of Computer Science
kcm161@scarletmail.rutgers.edu

Linqi Xiao
Rutgers University
Department of Electricial and Computer Engineering
lx130@scarletmail.rutgers.edu

Aishwarya Harpale
Rutgers University
Department of Computer Science
ach149@scarletmail.rutgers.edu

Neil Pillai
Rutgers University
Department of Computer Science
neil.pillai@rutgers.edu

Abstract

Most real-world problems are characterized by data simultaneously collected from several sensors. In machine learning, multi-view representation learning refers to the setting where the model learns the multiple views the data comes in. In this project, we apply this concept to cross-modal retrieval for Food AI [5, 7, 8, 9, 10]. We do this for both the im2recipe and recipe2im problems using the Recipe 1 Million[8, 10] dataset for cross-modal representation learning. In this final report on the project, we report our findings, and visualizations of the results for each of the three approaches used to tackle the aforementioned problems: 1) learning linear cross-modal representations using CCA [7], 2) learning non-linear cross-modal representations using feed-forward neural networks [7] and, 3) learning representations with cross-attention using transformers [6, 11].

1. Introduction

For quite some years at the start of the deep learning revolution, and before that the fields of computer vision and natural language processing seemed to be moving forward independently. However, with the advent of deep learning, it became easier than ever to learn and merge features belonging to the two fields viz. image and text, respectively. This area of cross-modal learning is a very interesting development in the field and has many use cases like cross-modal retrieval, translation and alignment.

A specific use case of cross-modal retrieval can be seen in Food AI. The problem statement can be expressed as: train machines that can automatically understand the type of food by analyzing the ingredients list, food images, and cooking instructions. This cross-modal representation learnt can then be extended to retrieval where, given an image, the model can return its corresponding ingredients or cooking instructions and vice versa.[10]

The following sections outline the prior work on this topic of cross-model retrieval for food, discuss the technical details of the approach followed by us, including the dataset and evaluation strategy used. We also discuss the results obtained from all the approaches followed and provide intuitive reasoning for the same.

2. Prior Work

The problem of cross-modal retrieval in the food domain, that is the so-called im2recipe problem, first introduced in [10], also introduced the Recipe 1 Million dataset. The authors of this paper postulate that the advancements in categorizing different types of food was made possible due to a larger dataset such as the Food-101 dataset [2], and follow along these lines of reasoning to introduce and establish a baseline using a multi-modal neural model on the Recipe 1 Million dataset. The multi-modal neural model that is trained uses word2vec and bidirectional LSTMs with skip-gram to encode the ingredients and instructions and ResNet-50 or VGG-16 to encode the corresponding recipe images. This paired encodings are then refined using semantic regularization and passed on to a CCA model that learns the joint space representations. The paper also provides visualization of the results in the form of retrieval examples, semantic vector arithmetic and attention maps on the text. The same authors expand on the dataset in [8] and also provide results for ablation studies and tSNE visualization of the learnt embeddings. We take inspiration from these papers for our stage 1.

Authors in [7] improve on the previous work done by the creators of the dataset by modifying the encoder architecture as well as the training procedure used, and add a food image synthesizing module conditioned by the textual information on top. They use WordPiece tokenizer followed by BERT to encode the text and ResNet-50 to encode the images. The output of these two is then passed through two and one fully connected layer, respectively before being projected onto a shared space. The training procedure is modified by using triplet loss with cosine similarity and hard sample mining as the objective function rather than the Euclidean distance that is typically used by CCA. We take inspiration from this paper and use the embeddings generated by the aforementioned architecture in stage 1 and 2.

Other approaches to learning cross-modal representations in the shared space have also been proposed. Examples include cross-modal kNN (C-kNN) [5]. Other use cases that leverage the Recipe 1 Million dataset include generating cooking instructions from the corresponding food image and ingredient list [9].

More recently, focus has shifted to analyzing and visualizing the ”why” behind the retrieval results: visualizing the attention between images and text for the recipe and vice versa. Modality Consistent Embedding Network [6] is a latent variable model that is able to learn modality-invariant cross-modal representations of the data, and also provide attention map visualizations of the objects in the images and the ingredients used in the recipe. The MultiModal Regularizer module in [11] uses two transformers that learn cross-attention between the image and text and vice versa, after hierarchically encoding the textual information using self-attending transformers and the image information using Vision Transformers. [4]. For learning and visualizing cross-attention in Stage 3 of our project, this is where we take inspiration from.

3. Technical Details

3.1. Preprocessing Data

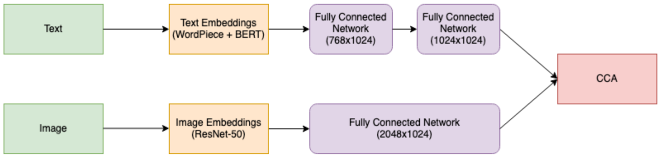
We first download the dataset provided on the official website of the im2recipe paper. Specifically, in accordance with the format in which the dataset was released [10], we download the *layer1.json* and *layer2.json* files. The *layer1.json* file contains information on the recipe images in the dataset: a recipe id, the image filenames for that recipe and the image download URLs. The *layer2.json* contains the corresponding text information for the recipe: corresponding recipe id, the title for the recipe, the ingredients used in and cooking instructions for the recipe. In addition, we also download the folder containing the images (given by the URLs) provided by the authors.

We preprocess the above data into its corresponding train, validation and test partitions, while making sure to align image filenames for the image data with the recipe instructions, ingredients and title for the text data using the recipe id. We store this preprocessed data and use it to train a model end-to-end in Stage 3 of our project.

3.2. Stage 1: CCA

For learning cross-modal representations using linear CCA, we use the architecture outlined in 1. Here, the text features for all of ingredients, instructions and title are extracted using pretrained WordPiece tokenizer and BERT [3], and the corresponding image features are extracted using pretrained ResNet-50. The extracted features are then passed to fully connected layers for further finetuning and to bring them down to the same dimension [8]. Next, both the image and text features (now brought to a same dimension size of 1024) from the training dataset are used to train a CCA model. We use this trained CCA model to then perform dimensional analysis on the validation dataset, and ablation studies and visualization on the test dataset, by transforming the extracted features to a shared latent embedding space using this CCA model.

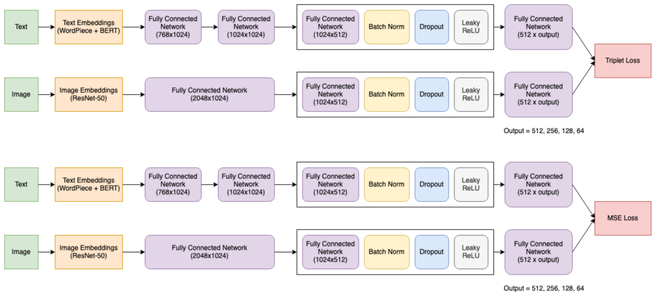
Figure 1. Architecture of the Stage 1


Text
Text Embeddings (WordPiece + BERT)
Fully Connected Network (768x1024)
Fully Connected Network (1024x1024)
CCA
Image
Image Embeddings (ResNet-50)
Fully Connected Network (2048x1024)

3.3. Stage 2: Non-linear Representations

In stage 2, we learn non-linear cross-modal representations by conceptually performing non-linear CCA using feedforward neural networks with non-linearity. As seen in 2, we use the features extracted in Stage 1 and then pass them separately to two instances of the same model, one each for textual features and image features. The model contains BatchNorm and Dropout layers for regularization and LeakyRelU layer for adding non-linearity to the representations learnt. The size of the output layer is varied as we perform dimensional analysis. We train these models using Mean Squared Error Loss and Triplet Loss. Training with MSE Loss can be thought of as being identical to learning a DeepCCA [1] model in the sense that we are using neural networks to bring the embeddings from two modalities down to a shared latent space. In models configured to learn using MSE Loss, we try to reduce the Euclidean distance between the image embeddings and text embeddings in the shared latent space. In models configured to learn using Triplet Loss, we perform random sampling for getting the negatives in the *(anchor, positive, negative)* triplets. Because we are using Triplet Loss, these models learn to bring the image and text embeddings for the same recipe nearer, and to push the image and text embeddings for different recipes, farther. We experiment with both Euclidean distance and cosine similarity as the similarity metric in models trained using Triplet Loss, and find that the ones with Euclidean distance perform better.

Figure 2. Architecture of the Stage 2

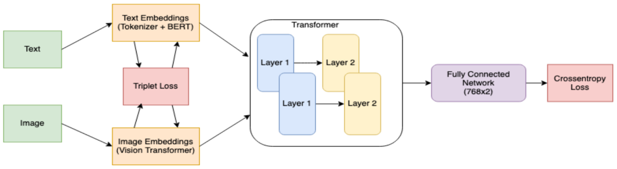

Text
Text Embeddings (WordPiece + BERT)
Fully Connected Network (768x1024)
Fully Connected Network (1024x1024)
Fully Connected Network (1024x512)
Batch Norm
Dropout
Leaky ReLU
Fully Connected Network (512 x output)
Triplet Loss
Image
Image Embeddings (ResNet-50)
Fully Connected Network (2048x1024)
Fully Connected Network (1024x512)
Batch Norm
Dropout
Leaky ReLU
Fully Connected Network (512 x output)
Output = 512, 256, 128, 64
Text
Text Embeddings (WordPiece + BERT)
Fully Connected Network (768x1024)
Fully Connected Network (1024x1024)
Fully Connected Network (1024x512)
Batch Norm
Dropout
Leaky ReLU
Fully Connected Network (512 x output)
MSE Loss
Image
Image Embeddings (ResNet-50)
Fully Connected Network (2048x1024)
Fully Connected Network (1024x512)
Batch Norm
Dropout
Leaky ReLU
Fully Connected Network (512 x output)
Output = 512, 256, 128, 64

3.4. Stage 3: Cross-Attention

Finetuning Extractors: For the final stage, we use models finetuned on the Recipe1M dataset to extract features. Because we want to learn the cross-modal attention between textual and image information for recipes, we need models that can tokenize both images and text, and apply attention on them. For tokenizing text and learning representations with self-attention, we already have BERT. For tokenizing and learning represenations with self-attention for images, we use the newly introduced Vision Transformers [4]. We finetune these two extractors by training the last two layers (and freezing the other layers) of both on the Recipe1M dataset using Triplet Loss with negatives generated using random sampling and with Euclidean distance as the similarity metric 3. We do this to ensure that better feature extraction takes place.

Learning Cross-Attention: We treat the actual retrieval problem as a binary classification problem where if the image and text embeddings are from the same recipe, the class is positive and otherwise it is negative. This is why we train the pipeline in 3 after finetuning the extractors using Cross Entropy Loss. To learn cross-attention between the extracted text and image features, we use a vanilla two-layer, two-head transformer encoder as a module, taking inspiration from [11]. To actually train the transformer encoder to learn cross-attention, we create a intermediary batch-wise temporary dataset which includes positive and negative classes in the ratio 1:2. This intermediary data is then passed to the transformer encoder which positionally encodes the image and text features before linearly projecting their CLS tokens which have been concatenated with each other with a SEP token in between to a two dimensional class probability. The class probability for the positive class is what we use for retrieval later.

Figure 3. Architecture of the Stage 3


Text
Image
Text Embeddings (Tokenizer + BERT)
Triplet Loss
Image Embeddings (Vision Transformer)
Transformer
Layer 1
Layer 2
Layer 1
Layer 2
Fully Connected Network (768x2)
Crossentropy Loss

4. Evaluation

4.1. Dataset

We use the Recipe 1 Million dataset [8, 10]. This dataset consists of 1 million text recipes that contain titles, instructions and ingredients in English. Additionally, a subset of 0.5 million recipes contain at least one image per recipe. Data is split in 281K train, 60K validation and 60K test image-recipe pairs, in accordance to the official data release provided in R1M. In addition, the recipe side of the data contains three components: title of the recipe, ingredients of the recipe and cooking instructions for the recipe.

4.2. Evaluation Strategy

We use the standard retrieval metrics as used in previous literature: medR and Recall@K for K = 1, 5, 10. We average these metrics over 10 runs on the validation data for dimensional analysis and test data for the ablation studies, each for a sample size of 1K and 10K, again in keeping with prior literature. For the third stage, instead of using cosine similarity between the image and text features as the means of calculating the ranks of the retrievals, we use the probability for the positive class, as explained in section 3.4.

A brief explanation about the metrics: **Recall Rate R@K:** This tells us how often the ground truth image or text retrieved places in the top K ranks, where for this project K = 1, 5, and 10.

MedR: MedR or median rank is defined as the median of ranks taken over the sample size where the rank is the position where the ground truth image or text was found to be during retrieval.

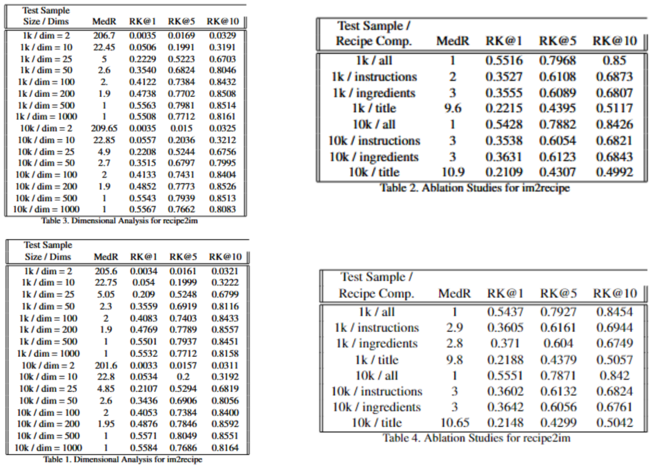
5. Discussion of Results

5.1. Stage 1: CCA

Figure 4 shows the results of dimensional analysis and ablation studies obtained in stage 1. Tables 1 and 3 in 4 show the dimensional analysis for recipe2image and image2recipe, respectively. As previously mentioned, the input embedding size of both image and text features into CCA is 1024. We transform these features to an output latent shared space size going from a maximum of 1000 down to 2. We can see that when latent dimension equals 500 and 1000, it performs better than other dimensions on all metrics, possibly because of less loss of information, as we reduce it from 1024.

Tables 2 and 4 in 4 show the ablation studies for image2recipe and recipe2image, respectively. We perform ablation studies using the best dimension that we found above (dim = 500). From these two tables we can see that when we use all the recipe components, we get better results since it contains the most information. We also see that when using either ingredients and instructions only as the textual information, the performance deteriorates, and using only the title performs the worst. Intuitively, this makes sense because the information in the title is only a few words. However, the title does differentiate between recipes where the ingredient and cooking instruction information cannot, and hence using all of it performs the best. We can also see that the performance of im2recipe and recipe2im is similar because we are learning the same shared embedding space, irrespective of the component order inside it. We also see that the performance for a larger sample size of 10k is somewhat worse than the sample size of 1k, simply because of more examples being seen.

Figure 4. Results of Stage 1: CCA



Test Sample Size / Dims	MedR	RK@1	RK@5	RK@10
1k / dim = 2	206.7	0.0035	0.0169	0.0329
1k / dim = 10	22.45	0.0506	0.1991	0.3191
1k / dim = 25	5	0.2229	0.5223	0.6703
1k / dim = 50	2.6	0.3540	0.6824	0.8046
1k / dim = 100	2.	0.4122	0.7384	0.8432
1k / dim = 200	1.9	0.4738	0.7702	0.8508
1k / dim = 500	1	0.5563	0.7981	0.8514
1k / dim = 1000	1	0.5508	0.7712	0.8161
10k / dim = 2	209.65	0.0035	0.015	0.0325
10k / dim = 10	22.85	0.0557	0.2036	0.3212
10k / dim = 25	4.9	0.2208	0.5244	0.6756
10k / dim = 50	2.7	0.3515	0.6797	0.7995
10k / dim = 100	2	0.4133	0.7431	0.8404
10k / dim = 200	1.9	0.4852	0.7773	0.8526
10k / dim = 500	1	0.5543	0.7939	0.8513
10k / dim = 1000	1	0.5567	0.7662	0.8083

Table 3. Dimensional Analysis for recipe2im

Test Sample / Recipe Comp.	MedR	RK@1	RK@5	RK@10
1k / all	1	0.5516	0.7968	0.85
1k / instructions	2	0.3527	0.6108	0.6873
1k / ingredients	3	0.3555	0.6089	0.6807
1k / title	9.6	0.2215	0.4395	0.5117
10k / all	1	0.5428	0.7882	0.8426
10k / instructions	3	0.3538	0.6054	0.6821
10k / ingredients	3	0.3631	0.6123	0.6843
10k / title	10.9	0.2109	0.4307	0.4992

Table 2. Ablation Studies for im2recipe

Test Sample Size / Dims	MedR	RK@1	RK@5	RK@10
1k / dim = 2	205.6	0.0034	0.0161	0.0321
1k / dim = 10	22.75	0.054	0.1999	0.3222
1k / dim = 25	5.05	0.209	0.5248	0.6799
1k / dim = 50	2.3	0.3559	0.6919	0.8116
1k / dim = 100	2	0.4083	0.7403	0.8433
1k / dim = 200	1.9	0.4769	0.7789	0.8557
1k / dim = 500	1	0.5501	0.7937	0.8451
1k / dim = 1000	1	0.5532	0.7712	0.8158
10k / dim = 2	201.6	0.0033	0.0157	0.0311
10k / dim = 10	22.8	0.0534	0.2	0.3192
10k / dim = 25	4.85	0.2107	0.5294	0.6819
10k / dim = 50	2.6	0.3436	0.6906	0.8056
10k / dim = 100	2	0.4053	0.7384	0.8400
10k / dim = 200	1.95	0.4876	0.7846	0.8592
10k / dim = 500	1	0.5571	0.8049	0.8551
10k / dim = 1000	1	0.5584	0.7686	0.8164

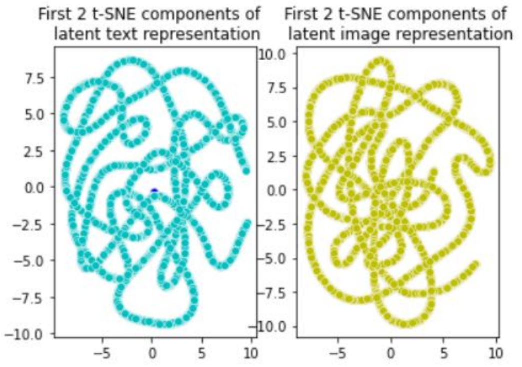
Table 1. Dimensional Analysis for im2recipe

Test Sample / Recipe Comp.	MedR	RK@1	RK@5	RK@10
1k / all	1	0.5437	0.7927	0.8454
1k / instructions	2.9	0.3605	0.6161	0.6944
1k / ingredients	2.8	0.371	0.604	0.6749
1k / title	9.8	0.2188	0.4379	0.5057
10k / all	1	0.5551	0.7871	0.842
10k / instructions	3	0.3602	0.6132	0.6824
10k / ingredients	3	0.3642	0.6056	0.6761
10k / title	10.65	0.2148	0.4299	0.5042

Table 4. Ablation Studies for recipe2im

The plot in Figure 5 represents text and image embeddings in the shared space (after CCA) for two types of recipes: muffins and cupcakes. On the left we plot the text embeddings and on the right we plot the image embeddings from the shared space. We can see that the text and image embeddings are similar in the points that they map to. The dark blue dot on the left in the centre represents the muffins embeddings and shows that different recipes map to different points in space and learn different manifolds. This successfully and intuitively shows the CCA learns embeddings properly.

Figure 5. Stage 1: CCA tSNE visualization of two different recipe types. Dimension = 500 [8]



5.2. Stage 2: Non-linear Representations

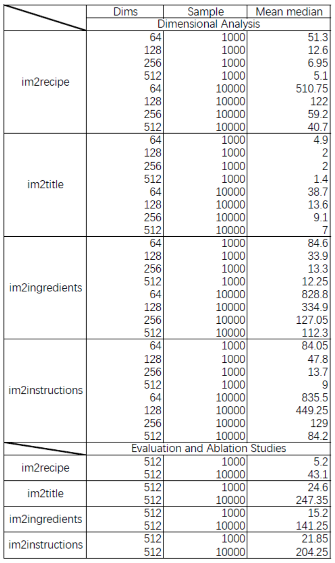
5.2.1 MSE Loss

Figure 6 shows the results for dimensional analysis and ablation studies for the non-linear shared representations learnt using mean squared error loss. We show the results for only one type of retrieval (from image to text) as the results for the reverse retrieval (from text to image) would be the same as seen in Stage 1, because we are learning shared representations, which do not change if the order of the modality inside them is changed.

From the results, we can see that we get the same pattern of results as in Stage 1. That is, as we decrease the shared latent space dimension size (the original input was of size 1024), the performance of the models decreases due to loss in information. Also similar to Stage 1, we see that the results when 10K samples are taken are worse than when 1K samples are taken. This is again due to more examples being seen, and the variability that comes with it. The pattern of retrieval results due to change in the textual information, however, is different than the one obtained in Stage 1. We see that models with image and title information (im2title) perform the best, followed by image and full text information (im2recipe), and lastly the worst are im2instructions and im2ingredients which are close in performance to each other. We argue that the im2title models are overfitting on the validation dataset, and achieving better performance because they have less number of tokens to close the Euclidean distance gap for since the words in title are less.

The first part of our argument above is proved correct when we perform ablation studies on the test dataset on the best performing dimension (dimension = 512). We see that im2recipe is the best performing, followed by im2ingredients and im2instructions, and im2title is the worst performing. Our argument here is that the im2title were overfitting and hence not able to generalize. In addition, the less information contained in titles also acts against

Figure 6. Results of Stage 2: MSE Loss

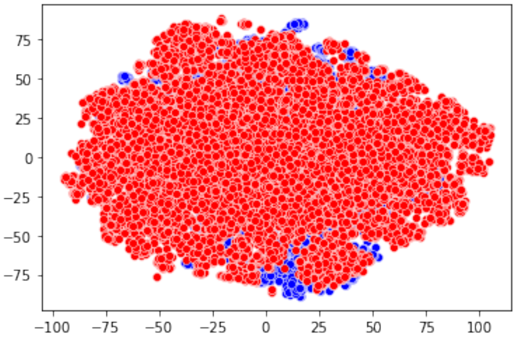


	Dims	Sample	Mean median
	Dimensional Analysis		
im2recipe	64	1000	51.3
	128	1000	12.6
	256	1000	6.95
	512	1000	5.1
	64	10000	510.75
	128	10000	122
	256	10000	59.2
	512	10000	40.7
im2title	64	1000	4.9
	128	1000	2
	256	1000	2
	512	1000	1.4
	64	10000	38.7
	128	10000	13.6
	256	10000	9.1
	512	10000	7
im2ingredients	64	1000	84.6
	128	1000	33.9
	256	1000	13.3
	512	1000	12.25
	64	10000	828.8
	128	10000	334.9
	256	10000	127.05
	512	10000	112.3
im2instructions	64	1000	84.05
	128	1000	47.8
	256	1000	13.7
	512	1000	9
	64	10000	835.5
	128	10000	449.25
	256	10000	129
	512	10000	84.2
	Evaluation and Ablation Studies		
im2recipe	512	1000	5.2
	512	10000	43.1
im2title	512	1000	24.6
	512	10000	247.35
im2ingredients	512	1000	15.2
	512	10000	141.25
im2instructions	512	1000	21.85
	512	10000	204.25

im2title models being the best performing, and we see a pattern in results similar to Stage 1.

Figure 7 shows a random subset of 10K image and text embeddings generated from the validation dataset using the im2recipe model with output dimensions = 512. We see that both the image and text shared embeddings that are generated map to almost similar points in space, thus showing that the Euclidean distance between image and text embeddings is successfully reduced.

Figure 7. Stage 2: MSE Loss tSNE visualization of shared latent space. Dimension = 512 [8]



5.2.2 Triplet Loss

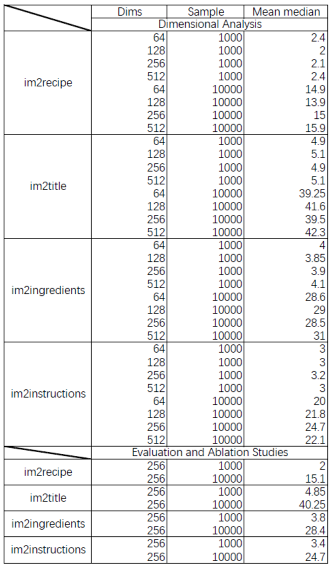
Figure 8 shows the results for dimensional analysis and ablation studies for the non-linear shared representations learnt using triplet loss. Again, we only show results for the retrieval from image to text.

These are the results for the three patterns we repeatedly analyze: 1) the performance seems to remain constant as we change the latent space dimension size; this is possibly because the way triplet loss pulls the positive anchor closer and pushes the negative farther away makes the model robust to changes in dimension size, 2) having the full textual information performs the best, followed by instructions and ingredients and then, the title, 3) the retrieval performance with more samples is noticeably worse than with fewer samples. We see that the dimension size of 256 performs marginally better than dimension size 512.

We then perform ablation studies taking the models with the best performance, that is with output dimension size 256. We see that we get the same pattern in results as earlier: having the full textual information is the best, followed by instructions and ingredients which are close to each other and rounded up by title.

However, the overall performance of training the same model architecture with triplet loss is far better than when the models are trained with MSE loss across all metrics as seen in Figure 9. We argue that this is because triplet loss works with a triplet and rearranges embeddings in space instead of just closing the gap between them. This results in every embedding being implicitly rearranged with respect to every other when using triplet loss, in comparison to it being rearranged with only the corresponding embedding in the other modality when using MSE Loss.

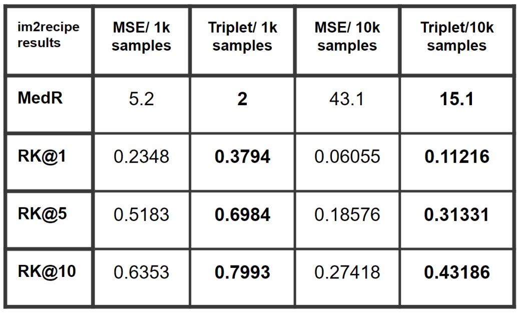
Figure 8. Results of Stage 2: Triplet Loss



	Dims	Sample	Mean median
	Dimensional Analysis		
im2recipe	64	1000	2.4
	128	1000	2
	256	1000	2.1
	512	1000	2.4
	64	10000	14.9
	128	10000	13.9
	256	10000	15
	512	10000	15.9
im2title	64	1000	4.9
	128	1000	5.1
	256	1000	4.9
	512	1000	5.1
	64	10000	39.25
	128	10000	41.6
	256	10000	39.5
	512	10000	42.3
im2ingredients	64	1000	4
	128	1000	3.85
	256	1000	3.9
	512	1000	4.1
	64	10000	28.6
	128	10000	29
	256	10000	28.5
	512	10000	31
im2instructions	64	1000	3
	128	1000	3
	256	1000	3.2
	512	1000	3
	64	10000	20
	128	10000	21.8
	256	10000	24.7
	512	10000	22.1
	Evaluation and Ablation Studies		
im2recipe	256	1000	2
	256	10000	15.1
im2title	256	1000	4.85
	256	10000	40.25
im2ingredients	256	1000	3.8
	256	10000	28.4
im2instructions	256	1000	3.4
	256	10000	24.7

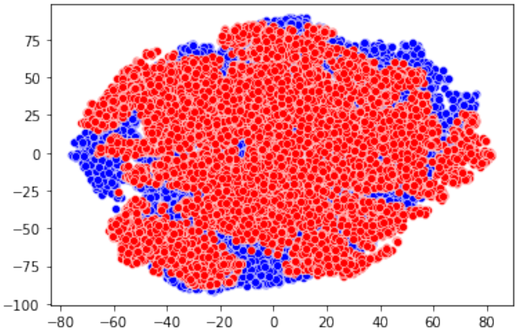
Figure 10 again shows random 10K image and text embeddings plotted in almost similar points in space, proving that the shared space is properly learnt, i.e., the image and text embeddings for the same recipe are points that are close in space, We note that this figure is generated using the im2recipe model for dimension 512 for comparison with the MSE Loss model of the same dimension, but which is not the best performing Triplet Loss model.

Figure 9. Comparison between MSE Loss and Triplet Loss. MSE Loss dimensions = 512. Triplet Loss dimensions = 256



im2recipe results	MSE/ 1k samples	Triplet/ 1k samples	MSE/ 10k samples	Triplet/10k samples
MedR	5.2	**2**	43.1	**15.1**
RK@1	0.2348	**0.3794**	0.06055	**0.11216**
RK@5	0.5183	**0.6984**	0.18576	**0.31331**
RK@10	0.6353	**0.7993**	0.27418	**0.43186**

Figure 10. Stage 2: Triplet Loss tSNE visualization of shared latent space. Dimension = 512



5.2.3 Semantic Vector Arithmetic Visualization

To showcase that the models are able to ”understand” the image and text representations semantically, we provide a vector arithmetic example for the im2recipe models trained using both Triplet Loss and MSE Loss. The dimensions for both are 512. From Figure 11, we can see that both the models learn the representations for the inputs (cake, cupcake and chocolate) and are able to retrieve these recipes properly. However, in this particular instance, the model trained using MSE Loss also outputs a correct vector arithmetic result (chocolate cake - cake + cupcake = chocolate cupcake) while the model trained with triplet loss does not.

Figure 11. Comparison between MSE Loss and Triplet Loss. MSE Loss dimensions = 512. Triplet Loss dimensions = 512 [10]

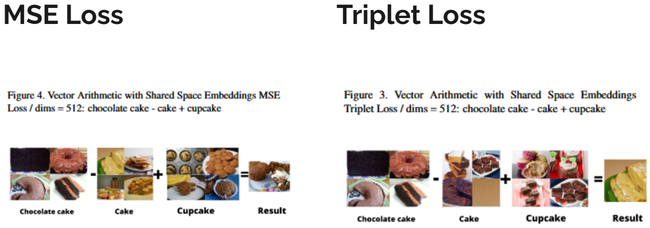


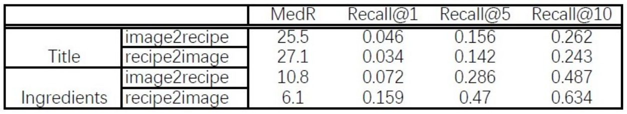
Figure 4. Vector Arithmetic with Shared Space Embeddings MSE Loss / dims = 512: chocolate cake - cake + cupcake

Figure 3. Vector Arithmetic with Shared Space Embeddings Triplet Loss / dims = 512: chocolate cake - cake + cupcake

5.3. Stage 3: Cross-Attention

Figure 12 shows the results for image to text and text to image retrieval using the cross-modal architecture. We perform ablation studies on only two components due to time restrictions. We can again see the same pattern in the results where the title information obtains the worst retrieval performance. This intuitively makes sense as previously explained.

Figure 12. Stage 3: Ablation Studies Results



		MedR	Recall@1	Recall@5	Recall@10
Title	image2recipe	25.5	0.046	0.156	0.262
	recipe2image	27.1	0.034	0.142	0.243
Ingredients	image2recipe	10.8	0.072	0.286	0.487
	recipe2image	6.1	0.159	0.47	0.634

5.4. Comparison of Results across Stages

We can see that CCA models serve as an effective baseline and achieve respectable performance in retrieval which deteriorates as we decrease the size of the shared space. These CCA models are able to beat the baseline results set forth in the original im2recipe paper [10] and comes close to the results in [7]. Models trained using triplet loss perform comparably to CCA models. We have used random sampling for the negatives and we argue that if hard sample mining is used, we can improve the performance of these models. The results using models with cross-attention are worse and we argue this is because we again use random sampling during finetuning with triplet loss. If we replace this with hard sample mining and train for more epochs, the performance should improve.

6. Conclusion

We tackled the Food AI problem by approaching the cross-modal retrieval tasks of im2recipe and recipe2im on the Recipe 1 Million dataset using incrementally better approaches. We learn 1) linear cross-modal representations using CCA, 2) non-linear cross-modal representations using feed-forward neural networks trained using mean squared error loss and triplet loss and, 3) representations with cross-attention using transformers. We perform dimensional analysis and ablation studies for all three of the above approaches and present their results. We also provide intuitive explanations behind the results and compare the results from all the three approaches. We find that models trained using triplet loss and random sampling of negatives perform the best for lower shared space dimensions, and that CCA performs best overall. CCA models also outperform the baseline results of the original im2recipe paper.

7. Group Contributions

The computation with respect to training, validation and testing of all the models was shared across all the team members' ilab accounts.

Kunj Mehta:

- Stage 1: Trained CCA models, performed dimensional analysis and ablation studies
- Stage 2: Experimented and researched for determining architecture, coded and trained models with MSE Loss, performed dimensional analysis and ablation studies for models with MSE Loss
- Stage 3: Wrote the code for and finetuned the BERT and ViT extractors on the Recipe1M dataset, wrote the code for and trained the cross-modal transformers

Linqi Xiao:

- Wrote the proposal for the project
- Stage 1: Wrote the report for Stage 1 of the project
- Stage 2: Wrote the report for Stage 2 of the project
- Stage 3: Wrote the report for Stage 3 of the project (the final report)

Aishwarya Harpale:

- Wrote the code for dataset preprocessing before Stage 1. (Section 3.1)
- Stage 1: Coded the functions for visualizations.
- Stage 2: Coded the functions for visualizations, including vector arithmetic.
- Stage 3: Researched and read papers to figure out the cross-attention part of the architecture.

Neil Pillai:

- Stage 1: Coded the function used for calculating the medR and recall metrics
- : Stage 2: Coded and trained models with Triplet Loss, performed dimensional analysis and ablation studies for models with Triplet Loss
- : Stage 3: Performed ablation studies

8. Project Code

The code for this project can be found in this GitHub repository. The files are: *Step1 + Step2.ipynb* which contains code for Stage 1 and 2, *Step-3.ipynb* which contains code for Stage 3, *preprocessing.ipynb* which contains code for preprocessing and aligning the original dataset and *Viz.ipynb* which contains code and visualizations shown in this report.

References

[1] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. *Proceedings of Machine Learning Research*, 2013. 2

[2] L. Bossard, M. Guillaumin, and L. Van Gool. Food-101–mining discriminative components with random forests. *European Conference on Computer Vision*, 2014. 1

[3] J. Devlin, M. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT 2019*, pages 4171–4186, 2019. 2

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 2, 3

[5] M. Fain, A. Ponikar, R. Fox, and D. Bollegala. Dividing and conquering cross-modal recipe retrieval: from nearest neighbours baselines to sota. *CoRR*, 2019. 1, 2

[6] H. Fu, R. Wu, C. Liu, and J. Sun. Mcen: Bridging cross-modal gap between cooking recipes and dish images with latent variable model. *CVPR*, 2020. 1, 2

[7] R. Guerrero, H. X. Pham, and V. Pavlovic. Cross-modal retrieval and synthesis (x-mrs): Closing the modality gap in shared subspace learning,. *Association for Computing Machinery*, pages 3192 – 3201, 2021. 1, 2, 6

[8] J. Marin, A. Biswas, F. Ofli, N. Hynes, A. Salvador, A. Aytar, I. Weber, and A. Torralba. Recipe1m+ : A dataset for learning cross-modal embeddings for cooking recipes and food images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 1, 2, 3, 4, 5

[9] A. Salvador, X. Drozdzal, G. Neito, and A. Romero. Inverse cooking: Recipe generation from food images. *CVPR*, 2019. 1, 2

[10] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Torralb. Learning cross-modal embeddings for cooking recipes and food image. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3020–3028, 2017. 1, 2, 3, 6

[11] M. Shukor, G. Couairon, A. Grechka, and M. Cord. Transformer decoders with multimodal regularization for cross-modal food retrieval. 2022. 1, 2, 3