Cross Modal Representation Learning

Dhruv Metha Ramesh
Rutgers University
Department of Computer Science
dhruv.metha@rutgers.edu

Anindita P Chavan
Rutgers University
Department of Computer Science
anindita.chavan@rutgers.edu

Jash Mitesh Gaglani
Rutgers University
Department of Computer Science
jash.gaglani@rutgers.edu

Sahil Rajendrakumar Raut
Rutgers University
Department of Computer Science
sahil.raut@rutgers.edu

1. Methodology

1.1. Dataset

We use the Recipe 1 Million Dataset [1, 2], which consists of approximately 1 Million text recipes with titles, instructions, and ingredients in English. We use both the **complete dataset** which contains 238999 train, 51119 validation, and 51303 test image-recipe pairs – a total of 800K (train + test + validation) images and a subset of approximately 0.5 Million recipes containing at least one image per recipe that contains about 400K (train + test + validation).

1.2. Prepossessing

The images dataset containing 800K images (train + validation + test) was downloaded from the im2recipe webpage. The webpage also has 2 other files – **layer1.json**, that contained the text recipe information and the key file **layer2.json** that contained the mappings between images and text.

We convert the text data into feature vectors of size **768** using the Bert Language Model. Bert does the WordPiece tokenization inherently and we extract feature vectors from the second last layer of the model. We extract features for 4 different types of text - **title, ingredients, instructions, and all of them concatenated**. We truncate the data at 512 token as that is the limit of the Bert transformer.

We use Resnet-50 to extract the data from the images. We remove the classification head and use model's last layer after pooling to get a feature vector for each image of size **2048**.

We also do analysis with the **features provided** with the assignment.

1.3. Model

We train a Canonical Correlation Analysis (CCA) model [3] on the extracted features of the training text-image pairs. CCA is used to find the latent space where the objective is to maximum the correlation between a linear combination of text features and linear combination of image features. Using CCA we were able to extract vector representations of the text and image data in the shared latent space for our test data set. We use the 'cca_zoo' library to make the model.

2. Evaluation

2.1. Metric

Our model described above is used to find an optimal shared latent dimension for both the image to recipe and recipe to image retrieval tasks. We evaluate these retrieval tasks of the CCA models using two benchmark metrics, namely median rank (medR)[4] and recall rate at top K (RK).

Median Rank medR: We compare the generated latent features of the images and text using "correlation" score for each with the other. The rank here is defined as the rank of the score the ground truth image latent vector when compared with it's text counterpart or vice-versa. Computing these ranks for the entire test set and taking the median is defined as the **medR** metric.

Recall Rate R@K: This is a standard retrieval metric, that tells us how often is the ground truth image retrieved in the top K ranks (as defined above), where for this project

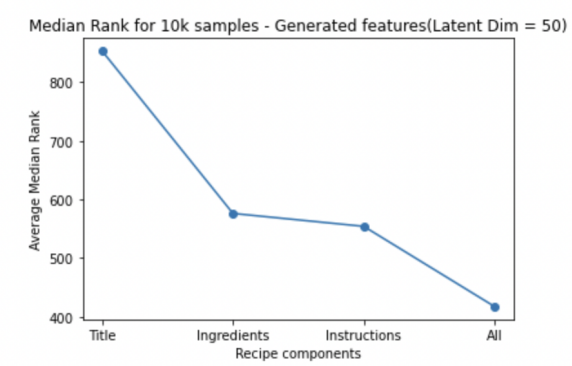

Figure 1. Median Rank for 10k samples - Generated features (dim=50)

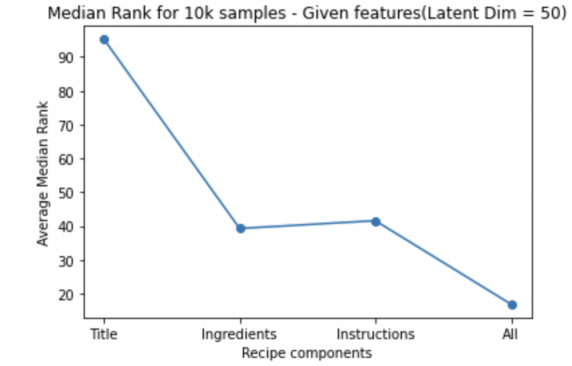

Figure 2. Median Rank for 10k samples - Given features (dim=50)

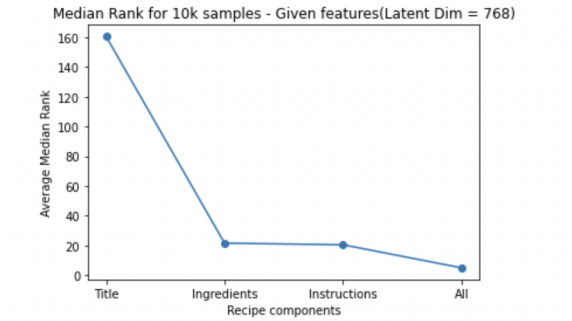

Figure 3. Median Rank for 10k samples - Given features (dim=768)

K = 1, 5, and 10.

2.2. Results and Ablation

Using the evaluation metrics given in Sec. 2.1, we can see the results in Table 1, 2 and Fig 1.

2.2.1 Generated Features

Using the raw features extracted from Bert and ResNet50 for the complete **training** dataset of **600k** recipe-image pairs (a recipe can have multiple images) and **test** dataset **130k** recipe-image pairs, we train the CCA model from cca_zoo and test it on our test set with the latent dimension set to **50**. We tried a bunch of latent dimensions (2, 25, 60, 100, 250, 300, 768), but we found 50 to be the best latent dimension. The ”best” was judged on the medR and R@K metrics.

	R@1	R@5	R@10
Generated Features(dim=50)			
Title	0.49	2.04	3.5
Ingredients	0.65	2.61	4.59
Instructions	0.58	2.56	4.52
All	0.69	3.09	5.46
Given Features(dim=50)			
Title	2.36	8.85	14.71
Ingredients	5.90	17.54	25.96
Instructions	6.5	18.75	27.47
All	10.52	29.04	40.67
Given Features(dim=768)			
Title	6.34	17.55	24.5
Ingredients	14.3	31.7	40.84
Instructions	15.24	32.84	41.36
All	27.75	52.9	62.99

Table 1. Recall for 10K sample size

	R@1	R@5	R@10
Generated Features(dim=50)			
Title	3.29	11.46	18.19
Ingredients	4.13	14.15	21.43
Instructions	4.38	15.15	22.39
All	**4.9**	**16.53**	**25.34**
Given Features(dim=50)			
Title	12.79	36.02	49.42
Ingredients	12.8	36.65	50.49
Instructions	23.29	52.5	65.85
All	34.9	68.6	80.4
Given Features(dim=768)			
Title	20.43	38.91	44.8
Ingredients	20.45	39.76	46.06
Instructions	35.58	57.66	64.24
All	**55.03**	**77.92**	**82.2**

Table 2. Recall for 1K sample size

Ablation From the results we can see that **instructions and ingredients find a lot more correlation with their images**, owing to the fact that the ingredients and instructions contain many pseudo labels for it's corresponding images. Like for example, using onions as a garnish, onions can be

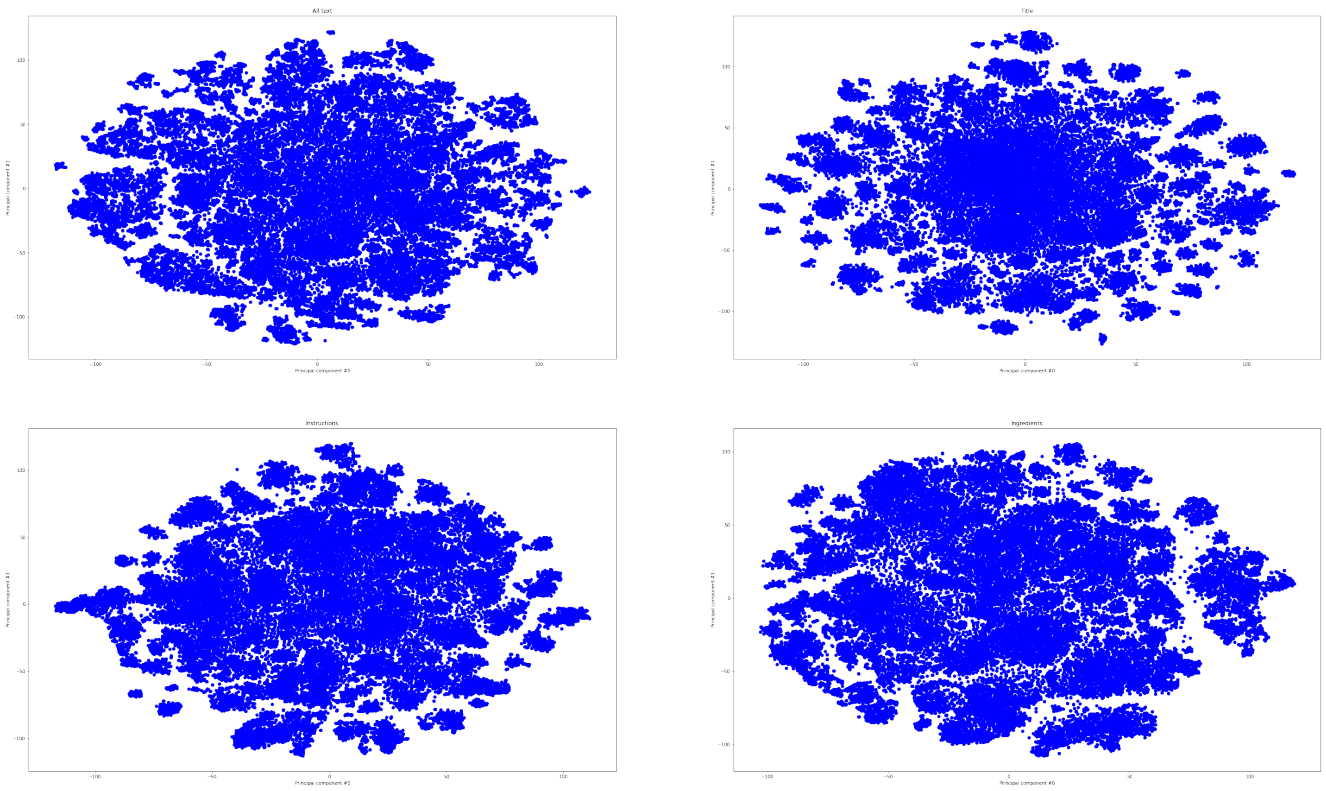

Figure 4. Median Rank for 10k samples - Given features (dim=50)

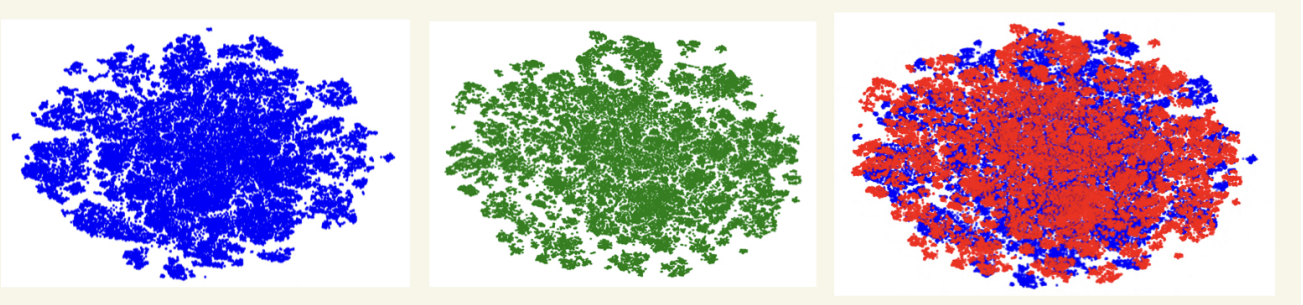

Figure 5. Median Rank for 10k samples - Given features (dim=50)

Figure 6. Median Rank for 10k samples - Given features (dim=50)

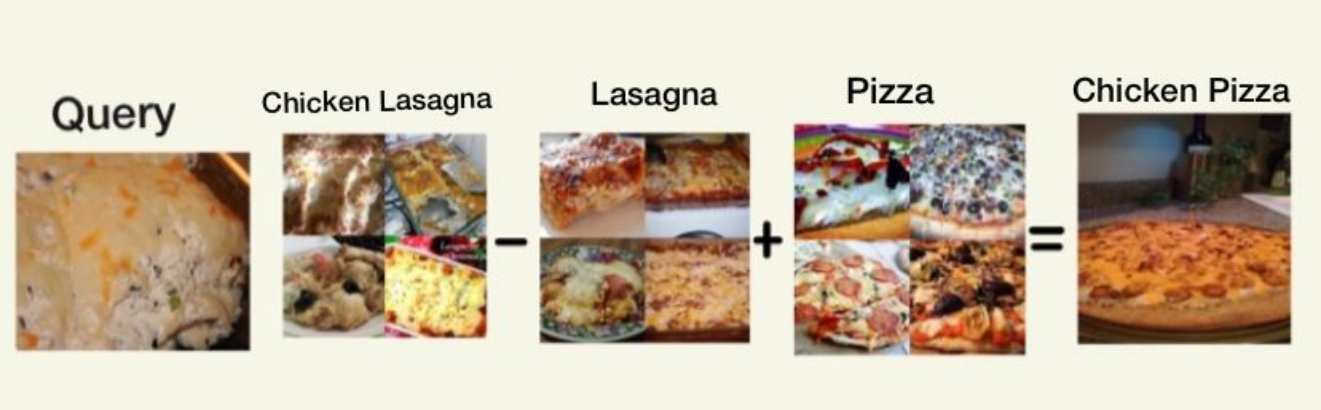

Figure 7. Median Rank for 10k samples - Given features (dim=50)

detected by the Resnet features and it's corresponding text "onion" in the recipe from the Bert features. CCA can learn this correspondence if we have multiple such recipe-image pairs in the dataset. This when compared to it's title text of "Indian Savoury Dish", does not contain this fine-grained information. The title talks about the more coarse information in the image.

We can conclude from this that **title** captures more **coarse** image features similar to image labels, but **ingredients and instructions** give us a more **fine** view into the image, for example the way something is cooked can define the color of the dish or the ingredients we had can influence the texture of the dish. Such representations are captured more for instruction and ingredients as compared to the title.

The **concatenated text** feature vector now contains **both the coarse and fine descriptors** (features) of the image, thus following the reasoning given above we see improved performances.

2.2.2 Given Features

The given features show significant improvement in the results, as (from our assumption) these features are fine-tuned non-linearly for the task of image to recipe and recipe to image. These results work very well with a higher dimension of the **latent space - 768** as compared to a smaller latent dimension. The reason is that these features (because they are fine-tuned) are very rich, and shrinking them to a smaller dimension causes loss of information.

Ablation The ablation for these features follow the same pattern as in Sec. 2.2.1. The features here however are richer and we thus see very good retrievals for the all the categories. The reasoning for the 4 text categories follows from the reasoning given above.

2.2.3 Visualization

We try to further make more inferences from the ablations for the 4 category text features using the t-sne plots in Fig 4. These plots contain all the 60k test-data points.

Ablation We see from the 4 plots in Fig 4:

1. **Title** we can see clusters but **not very clear boundary separations** between them in the shared space or it's very precise and this correlates with the image t-sne (which is not shown here, but looks exactly like this). The title gives very coarse information in the shared space thereby very mixed (a dish with eggs can have a variety of variants like pancakes, rolls, french toast, scrambled eggs etc) or very specific boundaries (similar names for similar recipes for cakes).

2. **Ingredients** we can see good cluster boundaries here but **bigger clusters**, indicating that a lot of "common" ingredient mapping to a variety of recipes.

3. **Instructions** Much **clearer boundaries** in comparison. This indicates the way a dish is used in the making of it determines a lot of how it looks.

4. **All** These form the best clusters of them all, giving us the best of all these relationships defined above.

Arithmetic Visualizations We use the given features and the latent features to recover some interesting arithmetics to show how strong the shared space features are.

3. Conclusion

For learning textual and image features from recipe, we used the Recipe 1M dataset. Canonical Correlation Analysis (CCA) model were trained to learn the cross model representation in latent space. Judging on the basis of medR and R@K metric, we determind 50 to be the optimal latent

dimension. From the ablation studies, we determined that the title gives us more coarse features while the instructions and ingredients gives us a fine view into the image.

References

[1] Y. A. J. M. F. O. I. W. A. Salvador, N. Hynes and A. Torralba. "learning cross-modal embeddings for cooking recipes and food images. *Proceedings of the IEEE conference on computer vision and pattern recognition*, page 3020–3028, 2017. 1

[2] F. O. N. H. A. S. Y. A. I. W. J. Mar´ın, A. Biswas 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

[3] K. P. Murphy. Probabilistic machine learning: An introduction. *TMIT Press,*, 2022. 1

[4] H. X. P. R. Guerrero and V. Pavlovic. Cross-modal retrieval and synthesis (x-mrs): Closing the modality gap in shared subspace learning. *New York, NY, USA: Association for Computing Machinery,*, 2021. 1