Project Report Step 2: Cross-Modal Representation Learning With Deep CCA and Triplet Loss

Govind Gangadhar
Department of Computer Science
gg676@scarletmail.rutgers.edu

Xin Li
Department of Electrical and Computer Engineering
xl598@scarletmail.rutgers.edu

Vincent Taylor
Department of Computer Science
vt152@scarletmail.rutgers.edu

Siddhant Mohan Kochrekar
Department of Computer Science
smk371@scarletmail.rutgers.edu

Abstract

Building upon linear canonical correlation analysis (linear-CCA) as proposed by Hotelling [1], to exploit the non-linearity of the features, an intuitive approach would be to use "kernel trick" to embed non-linearity, as proposed by [2], however it suffers computational issues with the included inner product and its nature of a non-parametric model [3]. To address this, we adopt Deep-CCA [3], which utilizes the multi-layer perceptron to efficiently computes the gradient of the correlation function. fixing the computational issue introduced by classical Kernel CCA

1. Approach

To further improve the model and increase the performance of queries on our embedding, we decided to to build upon our linear-CCA model using non-linear deep models and implementing triplet loss as a methods for evaluating cost. The goal of this approach is to see what our model and methodology gain beyond baseline linear-CCA. Our approach uses the same data-set as before referencing the Recipe 1 Million data-set. We will use the same evaluations metrics as our linear-CCA model, these include: Recall Rate for 10,5,and 1 as well as Median Rank. We will also conduct ablation studies to find the affects of hyper-parameters on our deep-CCA models and their cost functions.

2. Model

2.1. Configuration

We use two different loss functions to train two different pairs of models(image encoder and text encoder) to gauge the performance. Both the image and the text encoder models have the same configuration- two fully connected layers with 512 nodes each with regularizers being dropout probability of 0.3 and batch normalization. The input layer dimension is 1024 x 512 as we expect both the image and the text features to have 1024 feature dimension.

We use learning rate scheduler for Triplet loss based on the recall. The learning rate is 1×10^{-6} and weight decay is 1×10^{-7}. In both the cases (MSE and Triplet Loss), we pick the best model based by tracking the median rank of the validation set. This ensures us to have the best model for final test evaluation. We use Adam optimizer[4] to utilize the moment information for better convergence performance.

2.1.1 MSE-Loss

We have a deep model that learns the latent features of images and text and is optimized on mean squared error (MSE) loss.

2.1.2 Triplet-Loss

We have a second deep model that learns the latent features of images and text and is optimized on triplet loss (more details given below).

3. Dataset

We use Recipe1M dataset[23] which consists of around 800k RGB food images and over a million recipes in text. Each text data point contains an id, title, ingredients and instructions. From this, the image and text embeddings are extracted using pre-trained ResNet50 and BERT respectively. The images are in JPEG format The data is split into train, validation and test sets with 281598, 60422 and 60740 data points respectively. Both the image and text feature embeddings have a dimension of 1024.

4. Methodology and Background

4.1. Deep Canonical Correlation Analysis

Deep neural networks have seen increasing interests as the rapid development of computing and storage technologies of the recent years. A traditional method to handle nonlinearity in multi-view data is the Kernel CCA [2], which embeds a nonlinear kernel to the conventional CCA setting as proposed by [1], it can also introduce regularizations for high dimensional data which may result in singular covariance matrices [5] [2], however, the computation of the Kernel CCA is challenging as it introduces inner products, and it is a non-parametric model, which requires significant more time for training as the dataset grows large. To address this issue, we adopt the method of Deep Canonical Correlation Analysis (DCCA), which utilizes the power of modern deep neural networks to efficiently computes the gradient of the correlation to solve the correlation maximization problem efficiently using a stochastic gradient optimizer. With the Adam optimizer, we observe the DCCA model can learn the correlation efficiently by minimizing the mean-squared error and the triplet-loss [6], we can also see the model maximizes the cosine similarity and efficiently reduces the stochastic distance of the originally high dimensional data in a low-dimensional embedding using the tSNE metric [7].

4.2. Triplet-Loss

Triplet-loss [6] is a loss function that aims to minimize the distance between similar images and increase distance between contrasting images. Distance in this regard is measured by a distance function. The distance function primarily used in our model is Euclidean distance. Euclidean distance is defined as the the square root of the sum of the squared differences between the two vectors

$$\left\|\overline{a}-\overline{b}\right\|_2=\sqrt{\sum_{i=0}^{N}(\overline{a}_i-\overline{b}_i)^2} \tag{1}$$

where A and B are vectors. Triplet loss pairs three sets of data: Anchors, Positives, and Negatives. Positives and Anchors are the same class of object but distinct. Negatives are not of the same class, and as a result are also distinct of the anchor. We will use a to denote Anchor, p to denote Positive, and n to denote negatives.

Triplet loss for our model is calculated by finding the distance between the Anchor and Positive and the Anchor and Negative sets and making sure

$$\|f(\overline{a})-f(\overline{p})\|_2^2 \leq \|f(\overline{a})-f(\overline{n})\|_2^2 \Rightarrow \tag{2}$$

$$\|f(\overline{a})-f(\overline{p})\|_2^2 - \|f(\overline{a})-f(\overline{n})\|_2^2 + \alpha \leq 0 \tag{3}$$

we square the distances to remove the sign component associated with the distances and make everything scalar. However, there is one concern with the above equation and that is the trivial answer of setting all distances to zero to combat this we include a small margin to make sure our model is actually learning. we denote this with α.

$$L(a,p,n)=max(\|f(\overline{a})-f(\overline{p})\|_2^2-\|f(\overline{a})-f(\overline{n})\|_2^2+\alpha) \tag{4}$$

$$Cost=\sum_{i=0}^{N}L(A^i,P^i,C^i) \tag{5}$$

This results in our cost equation (5)

Generating Triplet sets is done in several ways the first way is random sampling. Using random.sample we find Anchor and then sample from a list of classes and ids that match that Anchor making sure we are not choosing the same Anchor for a positive. To find a negative we sample randomly from a list of classes and ids that do not match that of the anchors class. This makes simple, yet easy triplet sets. The classes describe are from the Recipe1M Validation pickle file. After these triplets are generated they are joined with their ids respective feature this allows us to generate triplets without having to have the data-frame loaded into memory. The number of random samples are a hyper parameter tuned by us. The number of triplet sets generated is around 1000 for our testing.

To create harder triplets for the model to learn from we approach the idea of creating close matching negatives. We generate Anchors and Positives the same way as before, except to generate the negative, we choose the sample that has the highest cosine similarity but belongs to a different label than the label of our Anchor and Positive.

5. Evaluation

5.1. Evaluation Strategy

For the evaluation strategy we will use the following retrieval metrics [8] [9] [10] [11]: Median rank which helps to determine independent rankings without the influence of dominant features. Each contribution is included regardless of the prominence or importance. The other method we will evaluate is the Recall Rate of the top K, which helps

to accurately measure what portion of actual positives were identified correctly within the top 1, 5, 10 . Where higher numbers mean more accurate retrieval between image-text and text-image queries. A baseline of random retrieval is established in all the tables below. Outputs of the evaluation data metrics are given below.

5.2. Evaluation Results

Following graphs depict the reduction in MSE loss $(Fig.1 - Fig.4)$ and Triplet loss $(Fig.5)$ at each epoch for the validation set for Recipe2Image task using deep-CCA.

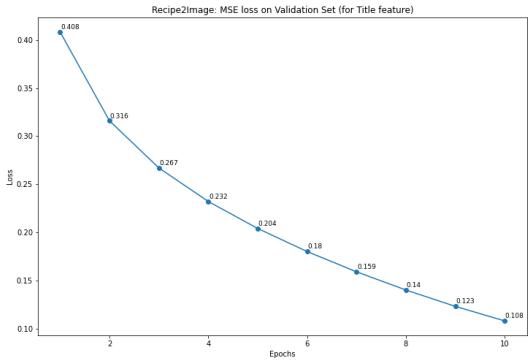

Figure 1. Training set MSE-loss for title features

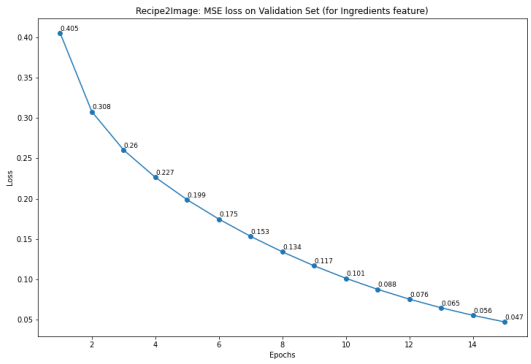

Figure 2. Training set MSE-loss for ingredients features

Embedding	Feature Set	R@1	R@5	R@10	MedRank
Recipe2Image	Title	3.15%	9.96%	15.77%	100
Recipe2Image	Ingredients	7.50%	21.28%	31.01%	31.7
Recipe2Image	Instructions	8.85%	23.99%	33.60%	28.05
Recipe2Image	All 3 Combined	21.05%	46.33%	58.14%	6.8

Table 1. Recipe2Image MSE Loss Evaluation Metrics (Test Set)

5.3. Evaluation Analysis

From the experiments, we can observe a consistent decay of the MSE for both the ingredient and the instruction

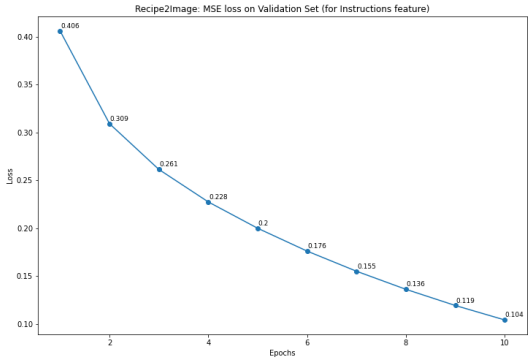

Figure 3. Training set MSE-loss for instructions features

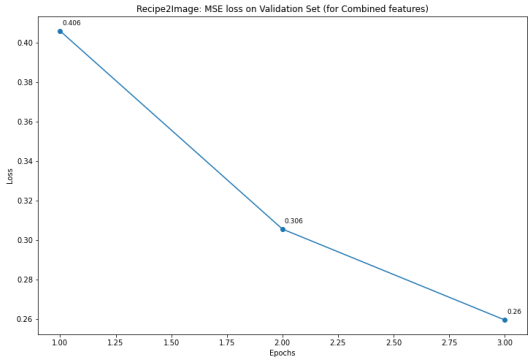

Figure 4. Training set MSE-loss for all three combined features

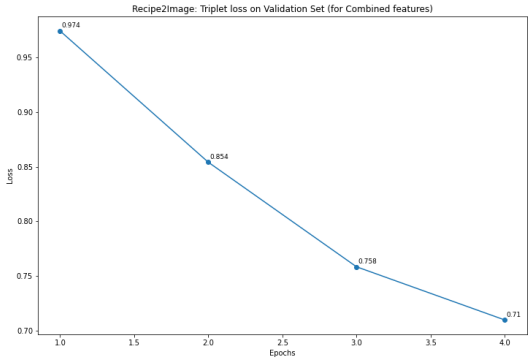

Figure 5. Training set Triplet-loss for all three combined features

Embedding	Feature Set	R@1	R@5	R@10	MedRank
Recipe2Image	Title	1.01%	3.16%	5.30%	327.1
Recipe2Image	Ingredients	14.1%	11.76%	40.63%	17.6
Recipe2Image	Instructions	15.8%	29.04%	44.60%	14.3
Recipe2Image	All 3 Combined	45.01%	72.07%	80.97%	2.0

Table 2. Recipe2Image Triplet-Loss Evaluation Metrics (Test Set)

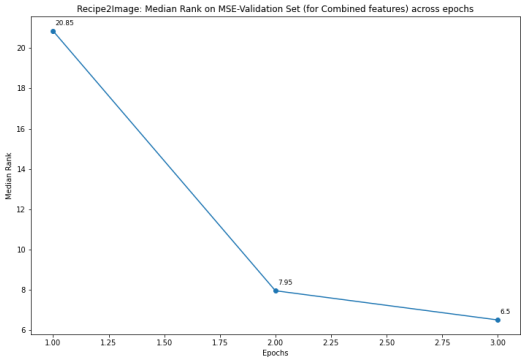

Figure 6. Median Rank from Validation set (MSE loss) for all three combined features

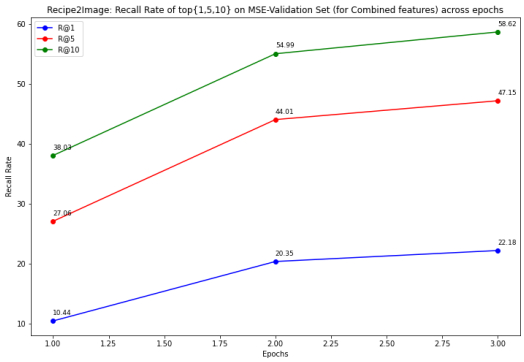

Figure 7. Recall Rate from Validation set (MSE loss) for all three combined features

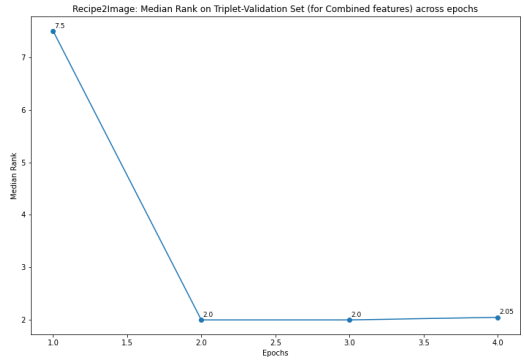

Figure 8. Median Rank from Validation set (Triplet loss) for all three combined features

features on the training set, which demonstrate the effectiveness of our Deep CCA model on extracting correlated pairs across different views of the data. Moreover, we can see the triplet-loss decay on the training set for the three combined

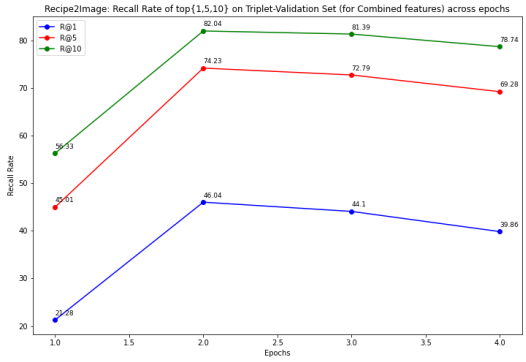

Figure 9. Recall Rate from Validation set (Triplet loss) for all three combined features

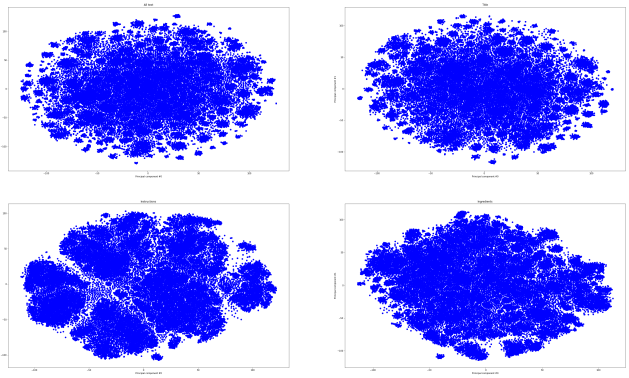

Figure 10. t-SNE on Textual Features using MSE Loss

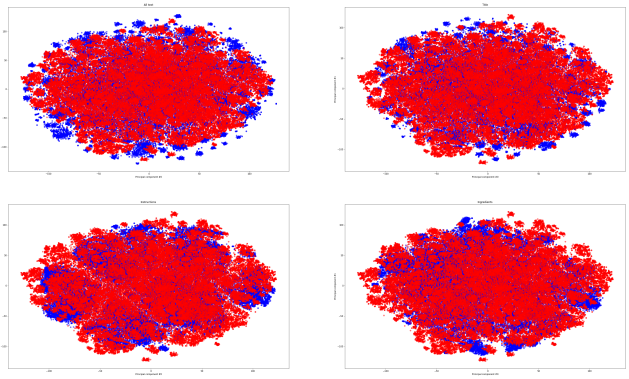

Figure 11. t-SNE on Image and Textual Features using MSE Loss

features (title + ingredient + instructions). As we expected, using triplet loss helps in improving the median rank from 6.8 for MSE loss to 2.0 for Triplet loss clearly indicating the effect of employing hard negative sampling. For the ranking metrics as proposed originally in [8], we have using the top-K recall rate (R@K), which defines the rate of successful retrieval (the percentage of queries for which the retrieval task successfully fetches the results among the top K candidates), along with the median rank, which defines

the median position of the retrieval response in the resulting list ordered by rank. The model performs well on retrieval tasks for both ingredient and instruction features and for all the three combined features on both performance metrics (MSE and Triple-Loss), while the triplet-loss metrics generally lead to a lower performance, showing it's a much stricter performance metric for retrieval tasks. For the tSNE [7] metric, which describes the stochastic distance of high-dimensional vectors in a low-dimensional embedding setting, we can see our model performs well in terms of aligning the text and image features by maximizing the correlation using deep neural nets. tSNE shows proper formation of clusters even on the MSE loss. However, it has the best cluster formation for the Triplet loss error-based model.

References

[1] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28(3/4):321–377, 1936. 1, 2

[2] Francis R. Bach and Michael I. Jordan. Kernel independent component analysis. *J. Mach. Learn. Res.*, 3(null):1–48, mar 2003. 1, 2

[3] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1247–1255, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. 1

[4] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. 1

[5] Xiangyu Wang, David Dunson, and Chenlei Leng. No penalty no tears: Least squares in high-dimensional linear models. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1814–1822. JMLR.org, 2016. 2

[6] Ricardo Guerrero, Hai X. Pham, and Vladimir Pavlovic. Cross-modal retrieval and synthesis (x-mrs): Closing the modality gap in shared subspace learning. In *MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia*, MM 2021 - Proceedings of the 29th ACM International Conference on Multimedia, pages 3192–3201. Association for Computing Machinery, Inc, October 2021. Publisher Copyright: © 2021 ACM.; 29th ACM International Conference on Multimedia, MM 2021 ; Conference date: 20-10-2021 Through 24-10-2021. 2

[7] Geoffrey E Hinton and Sam Roweis. Stochastic neighbor embedding. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT Press, 2002. 2, 5

[8] Micah Hodosh, Peter Young, and Julia Hockenmaier. Framing image description as a ranking task: Data, models and evaluation metrics. *J. Artif. Int. Res.*, 47(1):853–899, may 2013. 2, 4

[9] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator, 2014. 2

[10] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions, 2014. 2

[11] Amaia Salvador, Nicholas Hynes, Yusuf Aytar, Javier Marin, Ferda Ofli, Ingmar Weber, and Antonio Torralba. Learning cross-modal embeddings for cooking recipes and food images. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3068–3076, July 2017. 2