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

Abstract

In this report we talk extensively about cross-modal representation learning and the different techniques we used to explore the same. We explore image-text cross modal representation using linear techniques like Canonical Correlation Analysis (CCA) and it's non linear variant Deep CCA with different kinds of losses - Mean Squared Loss and Triplet Loss. We then move onto to the main theme of our project, learning video-text representations and doing so using a proxy task of alignment using Drop-Dynamic Time Warping (Drop-DTW) on an instructional video dataset. We talk about our qualitative and quantitative results and discuss further.

1. Introduction

Cross Modal Representation Learning has been one of the most relevant topics in the context of the world today. We generate massive amounts of data every second, be it video-audio-text from Youtube videos, tags or captions accompanying images from social media websites like Instagram, Twitter. Understanding such data brings value in tasks like image retrieval, video segment retrieval, step localization, captioning images, etc. Being able to understand one modality from the other brings us one step closer to intelligence like we haven't seen before. We humans think of pictures when reading something, describe an image with a caption in our brain, pick up words from audio, etc. For artificial neural networks like CLIP [11] and DALL-E [13] to be able to learn and represent such cross-modalities brings us very close to having a more unified learning space. In this paper, we explore techniques that bring these different modalities into a joint space. One such technique that would help us do this is Canonical Correlation Analysis (CCA) [8]. We explore both it's linear and non-linear versions, and validate it on how well it performs on a downstream retrieval task.

As mentioned before, with the large library of videos on Youtube, searching for something very specific can become very difficult with just the meta information – name, creator and tags. Combining the content information with language used for looking for that content, or in other words, giving the video feed a semantic meaning provides for an easier search for what we need. This is where we can use cross-modal representation learning between segments of videos and some text describing that segment. However, creating these representations using powerful neural networks require well labelled datasets – framewise (or a bunch of frames) description. This is a bottleneck as it is very expensive to create. To avoid such expensive operations, there are proxy techniques to learn these representations using self or weakly supervised learning techniques. In our case, we learn these representations using sequence to sequence alignment of instructional videos. Sequence to sequence alignment is where we use ordered time steps of both the video and text and try to align them with each other. Dynamic Time Warping (DTW) [15] is used extensively for the alignment problem, although very robust, it enforces correspondences between every segment of both sequences. In this paper, we explore the Drop-DTW [9] algorithm. This algorithm relaxes the enforced condition of finding alignments for every segment in DTW, by allowing us "drop" alignments that are not strong enough. Strong here is a measure that we can define based on a fixed threshold, or we can learn it using a learning algorithm. Main highlight of the report: Exploring the Drop-DTW algorithm for step localization using a learnable dropping function for learning cross-modal representations.

2. Related Work

Sequence alignment is a commonly arising problem in multiple fields related to machine learning, such as signal, bio-informatics, speech and audio analysis etc. It's applications range from human action recognition to temporal alignment of sequences. DTW, one of the most prominent methods in temporal alignment shares similarities with Soft-DTW [5] which takes advantage of a smoothed formulation of DTW along with a differentiable loss function to compute a soft-DTW score. This score considers the soft-minimum of the distribution of all costs across all possible alignments between two time series. A few other methods like the Deep Canonical Time Warping [16] combines CCA with DTW. This gives it the capability to handle observations of different or higher dimensions since it performs CCA to bring the sequences in a common latent subspace where they are maximally correlated. One of the major disadvantages of such DTW based approaches is that they enforce correspondences between all elements in both the sequences. Thus, it will result in a bad representation for sequences containing outliers. Drop-DTW helps overcome this drawback by providing the ability to skip outliers.

Representation learning focuses on learning compact numerical representations of various sources of signals like video, text, audio etc. The most common downstream tasks include video captioning, temporal action localization and text-to-video retrieval. One of the major drawbacks of using supervised approaches for representation learning is the requirement of dense labelling, especially in the case of large sized videos. Annotating the entire video with exact start and end time of each action can be computationally expensive. To overcome this, a few methods like the Discriminative Differentiable DTW (D3TW) [4] use sequence alignment as a proxy task. This weakly supervised approach learns to temporally align and segment video frames by leveraging the order in which the actions occur. Recent work in representation learning jointly models visual and audio components and works effectively on downstream tasks like action recognition. For our step-localization implementation, we use the MIL-NCE model [2] that leverages joint video and text representations to generate strong video and text embeddings.

Step localization in videos is important in information retrieval systems. Its most common application is temporal step localization that tries to find the precise start and end time for each action instance. Traditional methods in temporal step localization utilize sliding windows for sampling video segments and then compare sentences with each video segment individually to calculate matching relationships. This method, however, fails to achieve precise alignment between video and sentences. Fully supervised approaches provide a much better performance but have to rely on fine-grained temporal labels that indicate start and end times of each action instance. This can be cumbersome and expensive. Recent methods like NeuralNetwork-Viterbi[3] use weakly supervised approaches by using transcripts that provide an ordered list of actions for each training video. In our implementation of the Drop-DTW algorithm, we rely on the order of the instructional steps to provide a weakly supervised signal to our model.

3. Concepts

3.1. DTW

Dynamic Time Warping is an algorithm that uses Dynamic Programming to minimize the cost of aligning two sequences, there by obtaining optimal alignment. Let X and Z be the two sequences containing N and K lengths respectively, and they belong in $\mathbb{R}^d$. The DTW objective:

$$M^* = \underset{M \in \mathbb{M}}{\arg\min} \langle M, C \rangle = \underset{M \in \mathbb{M}}{\arg\min} \sum_{i,j} M_{i,j} C_{i,j} \quad (1)$$

where, $M_{i,j} = 1$ if Z_i aligns to X_j and 0 otherwise. $C_{i,j}$ is the **cost of matching** Z_i and X_j which can be any metric like cosine similarity or an inner product. Here $\langle M, C \rangle$ is the Frobenius inner product. This, however, enforces alignment between every Z_i and X_j. Even if they are outliers, DTW matches it to one of the features.

3.2. Drop-DTW

Drop-DTW allows for relaxing the constraint of aligning all sequence features by allowing to drop features that can be potential outliers. It does so by changing the objective to include a drop cost, which may be greater or lesser than the cost of matching. The new objective:

$$M^* = \underset{M \in \mathbb{M}}{\arg\min} \langle M, C \rangle + P_z(M).(d_i^z) + P_x(M).(d_j^x) \quad (2)$$

where $P_x(M)$ is a vector with the j-th element equal to one where the none of the X_j features match with the Z features. d_j^x is the cost of dropping the X_j.

This tells us that if it costs lesser to drop a feature than to match it, we should just drop that feature.
In our case, we are only interested in dropping columns (video features), so we have an objective that is:

$$M^* = \underset{M \in \mathbb{M}}{\arg\min} \langle M, C \rangle + P_x(M).(d_j^x) \quad (3)$$

We can further look at the algorithm 1.

```
Algorithm 1 Subsequence alignment with Drop-DTW.
Inputs: $C \in R^{KxK}$-pairwise match cost matrix, $d^x$ -drop costs for element in x
▷ initializing dynamic programming tables
$D^+_{0,0}$ = 0; $D^+_{i,0}$ = $\infty$; $D^+_{0,j}$ = $\infty$;    $i \in [[K]]$,$j \in [[N]]$    ▷ match table
$D^-_{0,0}$ = 0; $D^-_{i,0}$ = $\infty$; $D^-_{0,j}$ = $\sum_{k=1}^{j} d^x_k$;    $i \in [[K]]$,$j \in [[N]]$    ▷ drop table
$D_{0,0}$ = 0; $D_{i,0}$ = $D^-_{i,0}$; $D_{0,j}$ = $D^-_{0,j}$;    $i \in [[K]]$,$j \in [[N]]$    ▷ optimal solution table
for $i = 1, ....., K$ do    ▷ iterating over elements in Z
    for$j = 1, ....., K$ do    ▷ iterating over elements in X
        $D^+_{i,j}$ = $C_{i,j}$ + $min D_{i-1,j-1}, D_{i,j-1}, D^+_{i-1,j}$    ▷ consider matching zi to xj
        $D^-_{i,j}$ = $d^x_j$ + $D_{i,j-1}$    ▷ consider dropping xj
        $D_{i,j}$ = $min D^-_{i,j}, D^-_{i,j}$    ▷ select the optimal action
    end for
end for
$M^*$ = traceback(D)    ▷ compute the optimal alignment by tracing back the minimum cost path
Output: $D_{K,N}$, $M^*$
```

We see 3 matrix data structures that facilitate Drop-DTW which are:

1. D^+: This holds the cost of matching of that cell $C_{i,j}$ given whatever has happened before it.
2. D^-: This holds the cost of dropping a column (in our case specifically) given whatever has happened before it.
3. D: This holds the minimum of D^+ and D^-.

Since we have a non-differentiable min operator in the algorithm, we replace this with a **smoothmin**:

$$\text{smoothMin}(\boldsymbol{x}; \gamma) = \boldsymbol{x}.\text{softmax}(-\boldsymbol{x}/\gamma) \tag{4}$$

where γ is a temperature factor used to control how soft or smooth we want the minimizer to be. This makes the full process differentiable, and we can now back-propagate through this.

4. Cost Functions

Now, since our objective is to find good alignments between two sequences, using the Drop-DTW, we use the cost of matching, the final $D_{K,N}$ as mentioned in the previous section behaves as our ”cost” or ”loss”. However, there is an issue of degenerate alignment.

In our case, where we do asymmetric alignments, the video features of one sequence get aligned only around one or two of the most frequently getting aligned to step features. This degeneracy is avoided by adding a regularization function – a cluster loss as defined ahead:

$$L_{\text{clust}} = ||I - \hat{X}Z^T||_2 \tag{5}$$

where I is the identity matrix, $\hat{X} = (\hat{x}_1, \hat{x}_2, .., \hat{x}_K)$ such that

$$\hat{x}_i = \sum_{j=1}^{N} x_j.\text{softmax}(Xz_i/\gamma) \tag{6}$$

Equations 5 and 6 enable the model to be more dispersed. In equation 6, we do a form of attention pooling and see how much attention each frame pays to each step, and by minimizing the cluster loss, we can see that the frames are forced to pay attention to all of the frames with some temperature γ.

5. Dataset

We are using two different Datasets for the 3 steps of the project.

Step 1 and 2: Retrieval Task

We use the Recipe 1 Million Dataset [6], which consists of approximately 1 Million text recipes with titles, instructions, and ingredients in English. We used 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).

Step 3: Step Localization Task

We use the YouCook2 dataset [7] which contains 2000 long untrimmed videos from 89 cooking recipes. The instructional steps for each video are annotated with time segments and have an attached English instruction with it. We do a 67% : 23% split of the data for train, and validation sets. This leads to 1333 videos for training and 457 videos for validation.

6. Pre-Processing

We used different techniques to extract features from the above mentioned datasets.

6.1. Recipe 1M

The images dataset containing 800K images (train + validation + test) was downloaded from the im2recipe webpage. 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 the analysis with the features provided with the assignment which proved to be better than the features we extracted.

6.2. YouCook2

We use strong pretrained vision and language models trained using the MIL-NCE [2] objective, and use embeddings extracted from it. The MIL-NCE model has been pre-trained on the HowTo100M dataset [1] using a noise contrastive loss. We use features from S3DG [14], which consists of frame features that are grouped in segments of 4 seconds. Additionally, we also generate fearures using the downloaded videos for frame segments of 32 frames (approx. 1.5 seconds). We further also extract frame features by sampling the raw frame at random rates depending on the video FPS using the CLIP model, to generate at max 500 video frame features. We experiment with these features using the Drop-DTW algorithm and discuss the results.

7. Models

7.1. Model for Step 1 and 2

Canonical Correlation Analysis (CCA) is used to find the latent space where the objective is to maximize the correlation between a linear combination of text features and linear combination of image features. We used CCA to extract vector representations of the text and image data in the shared latent space for our test data set. We extracted features using both linear and non-linear projections of the data into a joint space. The non-linear feature extraction was done in two different ways using two different objectives – Mean Squared Loss and Triplet Loss.

7.2. Model Architecture for Step 3

We begin by describing the data-flow and how the model works end to end. We can see the full architecture in Fig. 1.

Normalization We first normalize the video and text features to have approximately zero mean and unit variance. This is so that when we do any transformation to either modality, they are still comparable when moved into a joint space.

Dataloader Here we use a special technique of loading the data into the model. From the 89 classes of different food items that we have, we sample 8 classes at random. From those 8 classes which consist of 10-12 videos each, we sample 3 videos. Thus, we'd then have a batch size of 24 (8 classes times 3 videos per class). This sampling is required for the model to train well as creating diversity in the batches help the model generalize better.

Fully Connected Layers for Video Features We setup a 2 layer fully connected network with the ReLU activation function to transform the video features to move into the joint space where the text resides. This helps improve the video features to be more jointly aligned to the text (step) features.
Here, we do not modify the step features and try to transform the video features to be aligned with the text features. This is our motivation to solve the alignment problem – generating good cross-modal features.

Loss / Objective Function We use 2 loss functions that were discussed in Section 4. We use the Drop-DTW algorithm to generate a cost that behaves as a loss signal. We also add a regularization term to avoid de-generate alignments. These losses are combined in a weighted fashion to create a final loss with which we teach the model to learn to align.

Learn-able Drops We use a neural network to learn the cost of dropping a frame, and this avoids setting a hard threshold value that needs to be tuned manually. This is then used in the Drop-DTW algorithm to compute the cost.

Inference We then use the learned drop cost and the features from the model to run the vanilla Drop-DTW algorithm (given in algorithm 1.) to infer the alignments for which you can see the results in Sec. 8.2

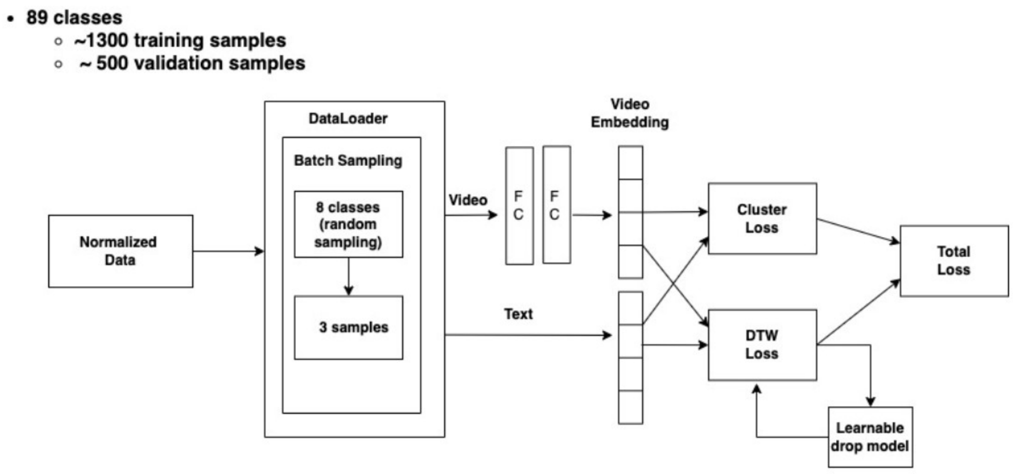

Figure 1. Step 3 Architecture

8. Evaluation

8.1. Metric for Step 3

The YouCook II dataset provides segment wise step annotations for each video along with other details like the duration and the total frame count. We use this data to calculate the ground truth label for each frame of the video. We use this labelled data to evaluate our model on the below metrics:

Framewise accuracy: It is calculated by taking the ratio of number of steps that have been correctly assigned to the ground truth label and the total number of steps. It gives us the percentage of correct frames when compared to the ground truth. [17].

Intersection over Union (IoU): It is calculated by taking a sum over the intersection between the predicted and ground truth time intervals divded by a sum over their unions. It is a stricter metric since penalizes the misalignment between video and text. [19]

8.2. Results for Step 3

Using the evaluation metrics given in Sec. 8.1, we can see the results in Table 1. We see improvements from the baseline model which has no learning component, just Drop-DTW. Adding a learnable video transformation along with a learnable drop cost, increases the Frame-Accuracy and IoU. On adding batch normalization and dropout, we see that we few get extra decimal points of accuracy increase. Since this is trained using a weakly supervised technique, using strong features and also enabling fine-tuning of the features using their orignal models can allow for even further increased accuracies.

We can see the two losses dropping in Fig 2. and Fig 3., ans also see the IoU and the frame accuracy increase in Fig. 5 and Fig. 6.

We also see some visual alignments in Fig. 7 and Fig. 8. We see almost comparable alignments, with DROPS in between frames indicating the presence of outliers. We can also see from Fig. 9 some degenerate alignments.

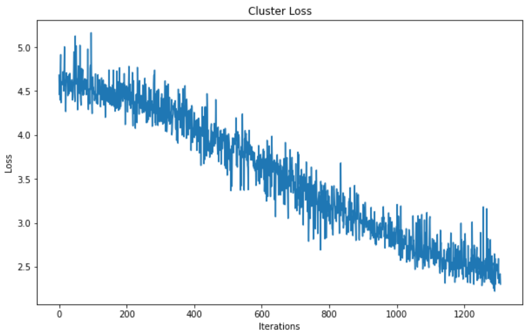

Figure 2. Epoch wise Cluster loss

8.3. Metric for Step 1 and 2

Our linear and non-linear CCA model is used to find an optimal shared latent dimension for the image and text features. The objective is to find a representation that maximizes the correlation between the two. We evaluate these models using two benchmark metrics namely the median rank (medR) [10] and the recall rate at top K. For calculating these metrics, we randomly sample 10K and 1K image-text pairs and sort them on the basis of their similarity. We run 10 such iterations and take a mean of the scores to obtain the final score.

	Frame Accuracy	Intersection Over Union
Baseline (no model)	55.78%	33.75%
2 Layer FC over video	58.02%	35.2%
2 Layer FC over video with batchnorm and droupout	**58.18%**	**35.75%**

Table 1. Frame Accuracy and Intersection over Union results

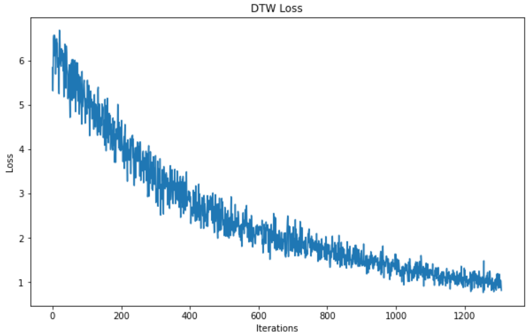

Figure 3. Epoch wise DTW loss

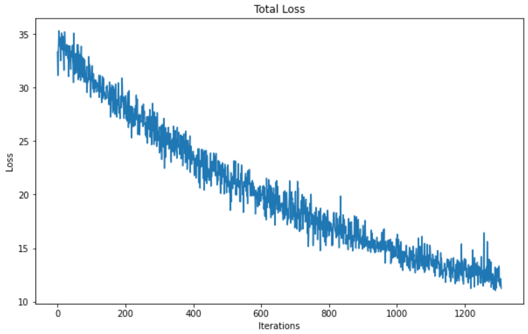

Figure 4. Epoch wise total loss

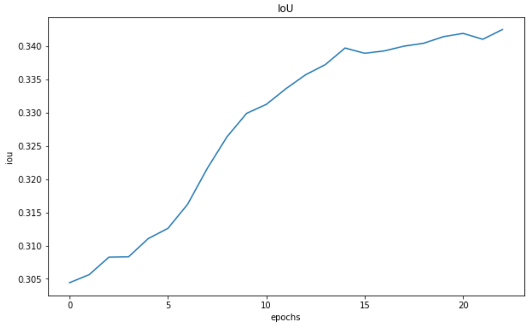

Figure 5. Epoch wise Intersection over union (IoU)

Median Rank medR: The median rank is calculated as the rank of the score of the ground truth image latent vector when compared with it's text counterpart or vice-versa, depending on whether we are performing an image to text retrieval or text to image. We record the rank in each iteration and calculate the median to get the median rank.

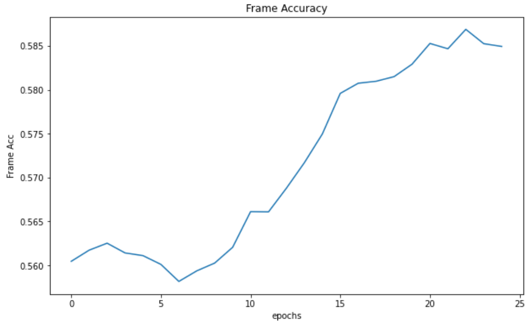

Figure 6. Epoch wise frame accuracy

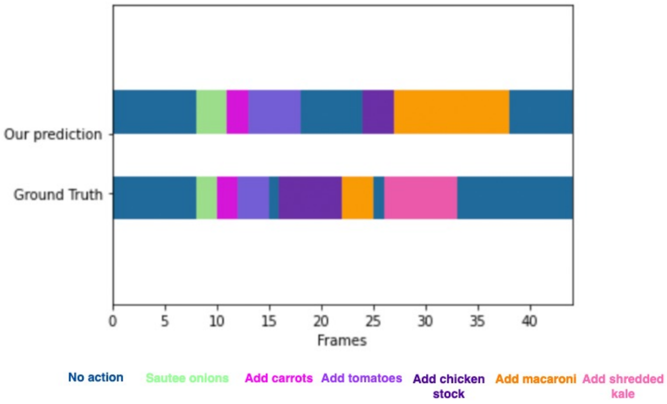

Figure 7. Video-Text alignment - Prediction vs Ground truth

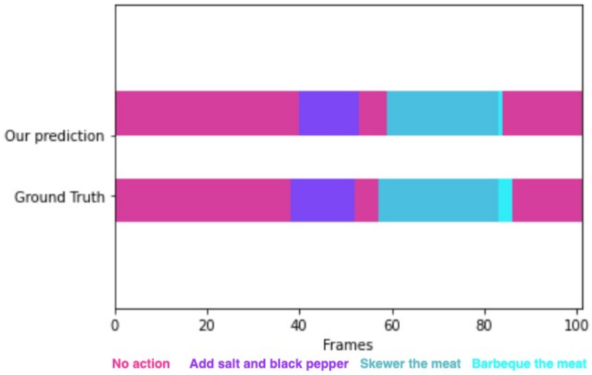

Figure 8. Video-Text alignment - Prediction vs Ground truth

Recall Rate R@K: This metric gives us an estimate of how frequently is the ground truth image retrieved in the top K ranks. For this project, we have estimated R@1, R@5

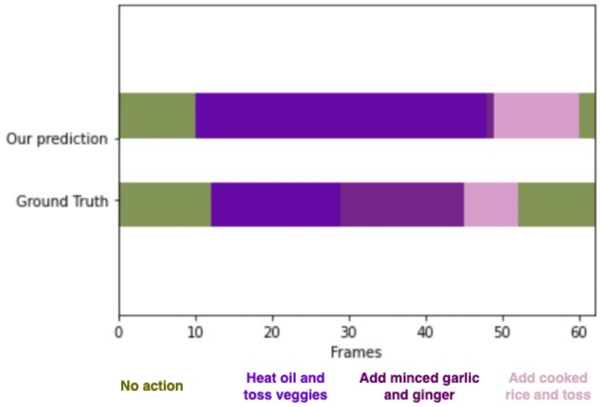

Figure 9. Video-Text alignment - Prediction vs Ground truth

and R@10.

8.4. Results and Ablation for Step 1

Using the evaluation metrics given in Sec. 8.3, we can see the results in Table 2.

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

8.5. Results and Ablation for Step 2

Using the evaluation metrics given in Sec. 8.3, we can see the results in Table 3 and Figure 10, 11, 12.

The triplet loss does far better in general on the **medR** and the **R@K** metric in comparison to the MSE loss because the triplet loss also takes into account the negative examples. This helps the data of a similar category be more clustered and different category away from such clusters, whereas in the MSE loss only the latter happens.

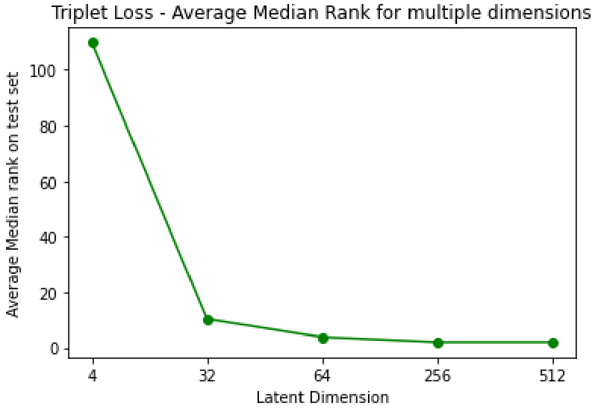

Figure 10. Triplet Loss - Dimension wise median rank

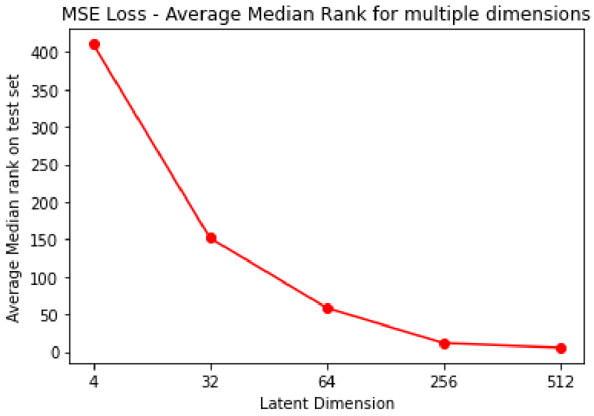

Figure 11. MSE Loss - Dimension wise median rank

8.5.1 Triplet Loss vs MSE

Ablation for title We can see clearly from the results that the titles behave strangely as it gives a better medR for MSE as compared to Triplet. This is not surprising as the title does not contain enough information about the features in the image, and finding negative samples for such may lead to choosing a positive sample as a negative one, something like a ”prawn curry” and ”shrimp curry”, they may be classified differently, but they are the same. The title feature vector holds very little semantic information of the detailed recipe, and hence pushing them away causes it to perform worse. We have performed a more detailed ablation study in the individual reports of step 1 & step 2.

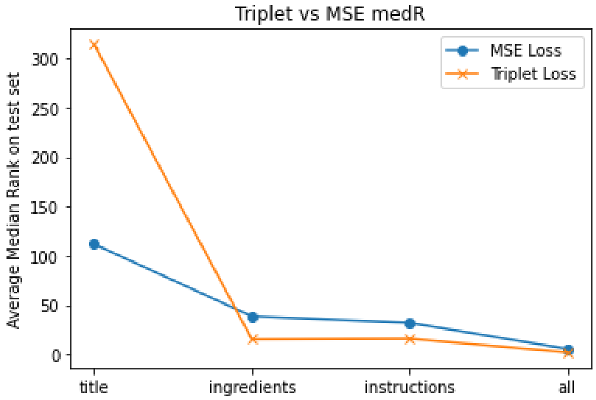

Figure 12. Triplet vs MSE medR

	medR	**R@1**	**R@5**	**R@10**
Generated(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	97.5	2.36	8.85	14.71
Ingredients	39.55	5.90	17.54	25.96
Instructions	42.2	6.5	18.75	27.47
All	11.5	10.52	29.04	40.67
Given Features (dim=768)				
Title	160.3	6.34	17.55	24.5
Ingredients	20.5	14.3	31.7	40.84
Instructions	19	15.24	32.84	41.36
All	7.5	27.75	52.9	62.99

Table 2. Recall and Median rank results for Phase 1

	medR	**R@1**	**R@5**	**R@10**
MSE Loss				
Title	111.7	2.81	9.29	15.07
Ingredients	38.55	6.54	18.98	27.57
Instructions	32.00	8.24	22.2	31.41
All	**5.55**	**24.14**	**49.39**	**60.3**
Triplet Loss				
Title	314.25	1.05	3.21	5.42
Ingredients	15.3	12.35	32.15	43.25
Instructions	15.95	12.58	32.28	42.79
All	**2.0**	**46.65**	**73.39**	**81.31**

Table 3. Recall and MSE results for Phase 2

9. Conclusion

Using the novel idea of **dropping** video frames that do not need to be aligned to steps, shows significant improvement over the standard DTW technique for sequence alignment. Since we do not use fine-grained labels to learn the representations and do so using weakly supervised learning, we also conclude the techniques like this can further lead to creating powerful representations in a very sample efficient way, removing the expense of labelling data.

Future Work With the popularity of large transformer models, it would be ideal to train an end to end cross-modal transformer similar to MMT [18] in a self supervised way with the MIL-NCE objective and an additional signal of alignment loss like the Drop-DTW loss. These multitask operations will help to generalize the model further, similar to the large language model T5 [12], but with multiple task heads.

We also see a bottleneck of requiring strong and connected vision and language features (or cross-modal features) which usually require extensive training. But once we have such models, fine-tuning them for cross-modal downstream tasks will bring out the real power of AI.

10. Project code

Github: `https://github.com/dhruvmetha/DropDTWPytorch/tree/main`

References

[1] J.-B. A. M. T. I. L. A. Miech, D. Zhukov and J. Sivic. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019. 4

[2] L. S. I. L. J. S. A. Miech, J.-B. Alayrac and A. Zisserman. End-to-end learning of visual representations from uncurated instructional videos. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),*, 2020. 2, 4

[3] A. I. A. Richard, H. Kuehne and J. Gall. Neuralnetwork-viterbi: A framework for weakly supervised video learning. *n Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR))*, 2018. 2

[4] Y. S. L. F.-F. C. Chang, D. Huang and J. C. Niebles. D3tw: Discriminative differentiable dynamic time warping for weakly supervised action alignment and segmentation.

International Conference on Machine Learning (ICML), 2017. 2

[5] M. Cuturi and M. Blondel. Soft-dtw: A differentiable loss function for time-series. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[6] 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. 3

[7] C. X. L. Zhou and J. J. Corso. Towards automatic learning of procedures from web instructional videos,. *AAAI Conference on Artificial Intelligence,*, 2018. 3

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

[9] K. G. D. A. G. A. D. J. Nikita Dvornik, Isma Hadji. Drop-dtw: Aligning common signal between sequences while dropping outliers. *https://arxiv.org/abs/2108.11996*, 2021. 1

[10] 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. 5

[11] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021. 1

[12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019. 8

[13] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. Zero-shot text-to-image generation. *CoRR*, abs/2102.12092, 2021. 1

[14] J. H. Z. T. S. Xie, C. Sun and K. Murphy. Rethinking spatiotemporal feature learning for video understanding. *https://arxiv.org/abs/1712.04851*, 2018. 4

[15] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken processing recognition. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1978. 1

[16] G. Trigeorgis, M. A. Nicolaou, S. Zafeiriou, and B. W. Schuller. Deep canonical time warping. pages 5110–5118, 2016. 2

[17] Y. R. Y. Z. D. Z. L. Z. J. L. Y. Tang, D. Ding and J. Zhou. Coin: A large-scale dataset for comprehensive instructional video analysis. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[18] S. Yao and X. Wan. Multimodal transformer for multimodal machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4346–4350, Online, July 2020. Association for Computational Linguistics. 8

[19] L. Zhou, C. Xu, and J. J. Corso. Procnets: Learning to segment procedures in untrimmed and unconstrained videos. *CoRR*, abs/1703.09788, 2017. 5