Reproducing Unseen Modality Interaction Results for the MSR-VTT Dataset

Odyssefs Drys-Pentzakis
odysseasdp@gmail.com
03119192

Gerasimos Markantonatos
gerasmark@hotmail.gr
03119149

Konstantinos Katsigiannis
kostas2001kat@gmail.com
03119127

Odyssefs Georgios Konias
odysseaskon@gmail.com
03119166

March 2024

The code for our reproduction can be found here.

Reproducibility Summary

Scope of Reproducibility

In this work, we analyze and try to reproduce the results of "Learning Unseen Modality Interaction" [1]. We focus on reproducing the results for the multimedia retrieval task, using the MSR-VTT dataset [2]. We attempt to see if these results can easily be replicated by following their methodology and trying to learn from modality-incomplete training data.

Methodology

We use the implementation of the model released by the authors for their paper. However, the code released by the authors is incomplete and not able to run at all. In addition, it only refers to one of the three datasets used in the paper (EPIC-KITCHENS), making the reproduction more difficult. The code is also missing important parts for the retrieval task such as the environments and dependencies used, the evaluation metrics and ablations studied in the paper.

Therefore, we had to we extend the code by following the details in the paper and adding the missing parts to fit the MSR-VTT dataset, which contains five more modalities than the EPIC-KITCHENS dataset, requires a different data loader, different evaluation metrics and tweaks in the model architecture and

loss function. We also found out that the model architecture was missing one layer specified in the paper that we had to add.

We conducted our experiments on an NVIDIA P100 GPU, requiring less than half an hour of training.

Results

The code released by the authors could not be run on the MSR-VTT dataset, so the reproduction was substantially different. There were many different errors on the code for which we had to implement our own solutions, which could differ from what the authors did.

Our results were close to the ones mentioned in the paper [1], and better than the papers the authors used for comparison. Our best implementation managed to get state-of-the-art results, close enough to the paper's claims, and sometimes even better.

Therefore, we conclude that the paper is somewhat replicable but definitely not readily reproducible, and the claims made by the authors are most likely valid. We can only conclude this for the multimedia retrieval part of the paper, and not the whole paper.

What was easy

It was easy to identify the main claims of the paper and understand the reasoning behind the model that the authors proposed. The code that the authors provided gave us a good idea of how to reproduce the results, although many steps were needed. We did not have to make substantial changes to the reorganization module and the alignment module of the code.

What was difficult

Setting up the development environment was challenging because of the lack of clear steps from the authors and a lack of a complete list of dependencies. Having to identify issues within the code and solve them ourselves was also very demanding, as we did not get any help from the researchers. The code having a missing layer in the transformer also took a lot of effort to fix.

We had to study other work on the same dataset to get an idea of how to proceed, mainly inspecting the works mentioned in the paper that are used as comparison, before we could start recreating the code. Most of our time was spent pinning down the dependencies and finding open source implementations of similar papers so that we could create our custom data loaders, metrics and loss functions.

Communication with original authors

We communicated with the authors via email, since they had not initially uploaded any code for the paper. They sent us their code for the EPIC-KITCHENS

dataset, which was supposed to be the code they submitted for review, but the code had many errors and could not be executed.

They claimed that it would be easy to reproduce the results, since the code was supposed to be the same for all three datasets they used, and all we would have to do it change the data loaders to load the MSR-VTT dataset. This was far from the truth, as most of the parts needed to execute the code for the MSR-VTT dataset were not included, and the parts that were included either had errors, or were incomplete (like the missing fully connected layer).

The authors did not provide any further help and did not respond to any more emails after the aforementioned claim.

Introduction

In the evolving landscape of artificial intelligence and machine learning, multimodal learning has emerged as a pivotal area of research. This field explores how machines can interpret and integrate information from multiple sensory modalities, such as visual and auditory inputs, to perform tasks akin to human perception. One significant application of this interdisciplinary research is in video classification and multimedia retrieval, where understanding and analyzing the interaction of various modalities - like images, text, and sound - are crucial.

The study "Learning Unseen Modality Interaction" by Zhang et al. [1] contributes to this domain by addressing the challenges of learning from modality-incomplete data and proposes a novel approach for generalizing to unseen modality combinations during inference. This method significantly impacts how we understand and process multimodal datasets.

Our research aims to reproduce and analyze the results of Zhang et al.'s study, specifically focusing on the MSR-VTT dataset, a comprehensive benchmark for evaluating multimedia retrieval systems. By replicating the study in the context of this particular dataset, we intend to validate the efficacy of the proposed approach in handling diverse and complex real-world data.

Methodology

The research done by Zhang et al. is extensive and covers three datasets, the EPIC-KITCHENS dataset [3], the robotics dataset by Lee et al.[4], and the MSR-VTT dataset [2]. Since our computational resources are far too limited compared to the resources used in [1] (the first dataset required three NVIDIA RTX A6000 GPUs), we will focus our efforts on the MSR-VTT dataset, which has the least computational requirements, used for multimedia retrieval.

The Dataset

We used the video-text retrieval MSR-VTT (Microsoft Research Video to Text) [2] dataset. The MSR-VTT dataset is a fundamental resource in video cap-

tioning research, each video being annotated with multiple human-generated descriptions, offering rich ground truth for the development and evaluation of video captioning algorithms. The dataset consists of 10K YouTube videos with 200K text descriptions. It consists of seven modalities: RGB video, audio, object, scene, face, OCR and speech.

As specified in [1], we directly used the unimodal features from [5] for our experiments. We loaded the features as a Kaggle dataset in order to access them freely. We inspected the pickle files from the github of [5], and managed to find the corresponding files for each modality. For the training of the model we used 1,702 samples with audio, OCR, speech and RGB and 4,811 video samples with RGB, object, scene and face. For validation, we used 127 video samples and for testing 765 video samples, both containing all the available modalities. Because of some modalities missing or being noisy, the RGB modality will be in both training sets.

The Model

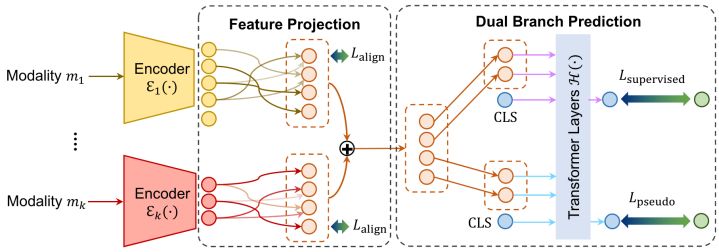

Figure 1: Method Overview

We attempted to build the model specified in the paper in order to replicate the researchers' experiments.

The model's input is seven different modalities. First, each modality is encoded by an unimodal encoder. In our case, we directly use the unimodal features Liu et al. [5]. Because these features are in different spaces and dimensions, a feature projection module is used to bring them in a common space. The feature projection module consist of six transformer layers [6], each with 8 heads and hidden dimension of 256. Then, feature alignment is applied with *Lalign* to each modality in training. A multimodal representation is obtained by adding all modalities available for the input sample together. To mitigate the risk of overfitting for specific modality combinations encountered in training, a dual-branch prediction framework is used. In this framework, features are categorized into two distinct groups, each corresponding to a separate branch of the model.

The first branch undergoes supervision using ground truth labels through *Lsupervised*, while the second branch is guided by pseudo-labels via *Lpseudo*. At the inference stage, the final prediction is derived by averaging the outputs from

both branches. The training objective for each data point is multifaceted, comprising three key components. Firstly, a feature alignment loss is incorporated, *Lalign*, which ensures that the features across all modalities align cohesively in a shared space, facilitated by our feature reorganization module. Secondly, a supervised loss, *Lsupervised*, is applied to the branch that leverages label annotations. The third component is a pseudo-supervised loss, *Lpseudo*, employed in the alternate branch that utilizes pseudo-labels.

We extended the code for the model that was provided by adding a fully connected layer for each modality, in order to get the output shapes we wanted.

The composite training objective is formulated as

$$L = \lambda Lalign + Lsupervised + \alpha Lpseudo,$$

where λ and α are hyperparameters balancing the loss terms. To enhance learning efficacy, we construct training batches by randomly selecting samples across all available training sets. Since we have more modalities now, we change the alignment loss to take into account all the available modalities.

For the MSR-VTT dataset, the researchers use the triplet margin loss function that requires a positive and negative sample for each sample. We instead extended the triplet loss for the whole batch and therefore implemented a Max Margin Ranking loss function, that can be used for our task. We managed to compute the supervised and pseudo loss with the help of this.

Alignement Module The model uses an alignment module and alignment loss in order to bring the unimodal features in a common space. To do that the authors introduce a series of learnable tokens which in our case are 128 vectors of dimension 256. For each modality the average feature $\overline{f}_m = \sum_m f_m^i$ is encouraged to be close to one of those tokens by incorporating its euclidian distance in the Lalign loss. In the retrieval task the selected token is the one closest to the average feature. So for each modality we calculate the euclidean distance betweenf_m and all the tokens and the feature alignment loss is:

$$L_{\text{align}} = \sum_{m \in M_1} \left\| \overline{f}_m - u_{nm} \right\|_2^2$$

where u_{nm} is the token with the smallest distance.

Pseudo Supervision The model uses a dual branching mechanism where two branches of outputs are produced. The first is compared to the groundtruth labels and the second to pseudo labels produced by the model. The authors argue that pseudo-supervision, when we have modality incomplete training data, helps the model focus on the more discriminative modalities. Specifically, certain modality combinations are more discriminative than others and training the model with the ground truth labels only will result in overfitting of the less discriminative modalities.

The dual branch is implemented inside the transformer and we get the two outputs with a single pass. However, neither the pseudo labels nor the code to reproduce them were given so we had to make them ourselves.

The authors compute pseudo labels for each modality by training the model with one modality at a time and averaging the output of the last 20 epochs. We tried a simpler approach, we calculated pseudo labels by training the model with one available modality at a time for 50 epochs and kept the output as the pseudo labels for this modality. In the end we have groups of modality-specific pseudo labels.

Since our training splits contain multiple modalities we have to choose which pseudo label we are gonna use for each sample. According to the paper: 'When M1 or M2 contain multiple modalities, we select the modality-specific pseudo-label closest to the groundtruth annotation with respect to the cosine similarity'. So we calculate the cosine similarity between each modality specific pseudo label and the corresponding groundtruth label and select the pseudo label with the highest similarity.

Experimental setup and code

The authors provided us with demo code to get us started. However, the code was not running, had mistakes, and there were no included instructions about the libraries used and their required versions. Also, it was dataset specific and could not be used with the MSR-VTT dataset.

Due to limited hardware resources, we decided to use Kaggle as a platform to run our experiments.

Our first task was to find the unimodal features and build the data loaders. Finding the correct features was made easy thanks to the detailed descriptions in [5], but loading them was very time-consuming due to the missing dependencies and the fact that we had to use a custom function for loading pickle files, found in [5].

The input for the model architecture had to be extended from two modalities to seven. Every modality had to be treated differently due to the different shapes of the files.

A mistake we found in the multimodal model named ViT was the lack of a fully connected layer in the end, causing problems and errors. So, we added it to the architecture. We found some lines of code that were never used and deleted them.

Next, we implemented a Max Margin Loss for the supervised and pseudo losses. The implementation was inspired by [7].

We chose to use the exact same metrics as in the paper, which are the most commonly used ones in multimedia retrieval tasks, and performed all the experiments done for the MSR-VTT dataset.

Results

Metrics

We use the same metrics that the authors use, and are commonly used in most multimedia retrieval tasks. These metrics include the recall at k (R@k) metrics, the median rank (MdR) and the mean rank (MnR).

The recall at k metric (**R@k**) measures the proportion of relevant items appearing within the top k ranked results. In our case, for example, when our input is a video and our output is a list of possible captions, the R@5 metric represents the number of the cases where a caption relevant to the input belongs in the first 5 captions of the output.

The median rank metric (**MdR**) checks the ranking of the first caption relevant to the input for each case, and computes the median ranking of those. Similarly, the mean rank metric (**MnR**) checks the ranking of the first caption relevant to the input for each case, and computes the mean ranking of those.

Implementation details

The number of learnable tokens for the feature alignment loss is set to $n_u = 128$ on MSR-VTT.

The length of the feature tokens after feature projection is set as $k^* = 16$ on MSR-VTT. The pseudo-labels are obtained by training the model for 50 epochs with one modality at a time and keeping the last prediction.

For the overall training objective the parameters are $\alpha = 1$ for retrieval and $\lambda = 0.001$.

Zhang et al. [1] used an NVIDIA RTX 2080Ti GPU with a batch size of 128 to train their model for multimedia retrieval. Their method was trained with 50 epochs and a learning rate of 10^{-2}.

We used an NVIDIA P100 through Kaggle for training and inference with batch size 128. The model was trained the same way as in the paper, for 50 epochs with a learning rate of 10^{-2}.

Results from the reproduction of the original paper

Method	**MnR ↓**
Late fusion	72.3
Modality Complete	
Gabeur et al. [8]	88.8
Nagrani et al. [9]	86.2
Wang et al. [10]	87.8
Modality Incomplete	
Shvetsova et al. [11]	72.3
Recasens et al. [12]	72.2
Unseen Modality Interaction	
This paper	66.2
Our results	**69.5**

Table 1: Comparison of Multimodal Learning Methods on MnR Metric for Multimedia Retrieval Task. The value we present is the mean MnR of Video to Text and Text to Video.

The approach is compared with recent multimodal learning methods, which assume all data to be modality-complete.

Table 1 shows the results for the multimedia retrieval task. Modality complete methods seem to perform worse than late fusion as they cannot learn the cross-modal correspondences without modality-complete data, and therefore cannot effectively make predictions for unseen modality combinations during inference. Existing approaches are not robust to unseen modality combinations, while this model obtains better generalization ability.

L_{align}	L_{pseudo}	**MnR ↓**
✗	✗	69.55
✓	✗	69.5
✓	✓	69.5

Table 2: Ablation of training objective using multimedia retrieval. We report the MnR Metric. In the first case we use only the supervised loss, and then we add each loss term. Both loss terms improve the generalization to unseen modality interactions.

In Table 1 and 2, the multimedia retrieval results are summarized with the Mean Rank (MnR) averaged between video-to-text and text-to-video. In Table 3 below, the full comparison is shown.

We can see that for the Video to Text task, our results are extremely close to the paper's, while for the Text to Video task our results differ a bit more, especially in the MnR and MdR metrics.

Below are the results for modality-incomplete testing, meaning that we limit the test set and do not use all the modalities, in order to see for which modal-

Method	Text to Video					Video to Text				
	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓	MnR ↓	R@1 ↑	R@5 ↑	R@10 ↑	MdR ↓	MnR ↓
Late fusion	5.2	16.9	27.7	32.0	72.1	5.4	17.6	26.5	31.0	72.4
Modality Complete										
Gabeur et al. [8]	3.5	15.8	26.1	35.0	76.5	2.5	11.2	18.6	47.0	101.1
Nagrani et al. [9]	3.8	16.1	25.6	35.0	75.2	2.8	12.1	18.6	46.0	97.2
Wang et al. [10]	3.5	16.1	25.9	35.0	76.6	3.1	11.6	19.0	46.0	98.9
Modality Incomplete										
Shvetsova et al. [11]	5.1	16.9	27.5	33.0	72.5	5.6	17.2	25.9	31.0	72.0
Recasens et al. [12]	5.0	17.1	28.2	33.0	73.2	5.2	17.4	26.5	30.0	71.2
Unseen Modality Interaction										
This paper	6.0	19.2	30.5	28.0	66.4	6.3	18.9	29.0	27.0	65.9
Our results	**3.9**	**14.4**	**25.1**	**37**	**77.8**	**7.2**	**21.2**	**30.7**	**25**	**61.2**

Table 3: Comparison with multimodal learning methods for the retrieval task using all metrics for MSR-VTT.

Model	Gabeur et al.	Nagrani et al.	Wang et al.	Shvetsova et al.	Recasens et al.	This paper	Our results
RGB, Audio, OCR, Speech	90.6	89.7	90.0	90.6	90.5	79.4	**79.9**
RGB, Object, Scene, Face	89.1	88.8	88.9	89.3	89.4	80.3	**74.3**
RGB, Object, Speech, OCR	92.4	90.2	91.5	78.1	77.4	70.2	**72.9**
RGB, Scene, Audio, OCR	91.0	89.9	90.7	75.3	74.2	70.3	**86.9**
RGB, Scene, Speech	95.6	94.5	95.1	80.2	79.3	74.3	**90.9**
RGB, Object, Audio	96.1	96.0	96.3	82.1	80.3	76.9	**73.2**
RGB, Speech	98.3	98.0	98.8	84.6	83.0	81.4	**89.4**
RGB, Audio	98.0	97.3	98.4	85.3	84.5	79.8	**88.5**

Table 4: Results for Modality-Incomplete Testing with multimedia retrieval.

ities our model performs the best. It becomes clear that some combinations of modalities perform far better than others, but none perform as good as the modality complete testing. This way, we can deduce which modalities may have lacking representations or may need more training.

Results beyond the original paper

This section emphasizes the course of work done from start to finish. To obtain the best results we had to go through a lot of approaches and slowly improve our model, data and hyperparameters.

Below we have summarized some of our results that show how utilizing L_{align}, L_{pseudo}, $L_{supervised}$, random labels, random OpenAI labels and a margin of approximately 0.1 in different combinations improve the results.

At first, we utilized the Word2Vec embeddings of the video captions. These embeddings are not as strong as the ones created by OpenAI. This is the reason why the results are significantly worse.

In addition, we utilized just the first caption of every video in the training phase. That was fixed after some trial and error, and after that captions were randomly picked every time to act as labels for the model.

Max Margin Loss has a margin parameter that was initialized to 1. After some testing we found that the model gives best results when $margin = 0.09381161988446174$.

The tests were done without the Object modality, and the last test was done with that modality included, to show how utilizing all the modalities also improves the results.

The below values are not entirely accurate as we did not run all the tests with

the same amount of epochs and all the other parameters constant. However, they are a good indication of how the results can change.

Method	Text to Video					Video to Text				
	R@1 ↑	**R@5 ↑**	**R@10 ↑**	**MdR ↓**	**MnR ↓**	**R@1 ↑**	**R@5 ↑**	**R@10 ↑**	**MdR ↓**	**MnR ↓**
Results without the Object modality and with Word2Vec embeddings										
$L_{supervised}$	2.7	10.6	19.0	44.0	97.1	3.0	11.9	20.4	51.0	111.6
$L_{supervised} + L_{align}$	2.5	8.8	15.6	69.0	141.0	3.4	11.6	18.0	52.0	122.7
$L_{supervised} + L_{pseudo}$	3.4	11.0	20.5	44.0	96.7	2.2	12.0	19.9	50.0	111.4
$L_{supervised}$ + random labels	2.6	9.9	18.7	46.0	99.6	3.9	12.7	20.4	41.0	99.3
Results with OpenAI word embeddings										
$L_{supervised}$	3.4	10.3	17.8	53.0	112.0	4.1	13.7	21.8	46.0	105.7
Results with margin = 0.09381161988446174 in Max Margin Loss										
$L_{supervised}$	3.7	11.9	20.7	42.0	91.9	5.5	18.3	28.1	30.0	71.3
$L_{supervised} + L_{align}$	3.7	11.9	20.7	43.0	91.7	5.5	17.9	28.4	30.0	71.2
$L_{supervised} + L_{align} + L_{pseudo}$	4.1	13.5	20.7	36.0	84.7	5.6	19.5	28.2	31.0	70.8
Results including the Object modality										
$L_{supervised}$	3.8	14.4	24.8	37.0	78.0	7.2	21.2	30.8	25	61.2
$L_{supervised} + L_{align}$	3.9	14.4	25.1	37.0	77.8	7.2	21.2	30.7	68.0	61.1
$\boldsymbol{L_{supervised} + L_{align} + L_{pseudo}}$	**3.9**	**14.4**	**25.1**	**37.0**	**77.8**	**7.2**	**21.2**	**30.7**	**68.0**	**61.1**

Table 5: How the results improve when utilizing different settings

Model	Best Validation Loss
Without the Object modality and with Word2Vec embeddings	
$L_{supervised}$	0.81635
$L_{supervised} + L_{align}$	0.85370
$L_{supervised} + L_{pseudo}$	0.81740
$L_{supervised}$ + random labels	0.81582
Without the Object modality and with OpenAI word embeddings	
$L_{supervised}$	0.86306
No Object modality, OpenAI word embeddings, and margin = 0.09381161988446174 in Max Margin Loss	
$L_{supervised}$	0.03814
$L_{supervised} + L_{align}$	0.03528
$L_{supervised} + L_{align} + L_{pseudo}$	0.03372
With Object modality included	
$L_{supervised}$	0.03111
$L_{supervised} + L_{align}$	0.03108
$\boldsymbol{L_{supervised} + L_{align} + L_{pseudo}}$	**0.03108**

Table 6: Best validation loss for each combination

References

[1] Yunhua Zhang, Hazel Doughty, and Cees G. M. Snoek. Learning unseen modality interaction, 2023.

[2] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[3] Dima Aldamen, Davide Moltisanti, Evangelos Kazakos, Hazel Doughty, Jonathan Munro, William Price, Michael Wray, Tobias Perrett, and Jian Ma. Epic-kitchens-100, 2020.

[4] Michelle A. Lee, Yuke Zhu, Peter Zachares, Matthew Tan, Krishnan Srinivasan, Silvio Savarese, Li Fei-Fei, Animesh Garg, and Jeannette Bohg. Making sense of vision and touch: Learning multimodal representations for contact-rich tasks, 2019.

[5] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts, 2020.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[7] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a Text-Video Embedding from Incomplete and Heterogeneous Data. *arXiv:1804.02516*, 2018.

[8] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval, 2020.

[9] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion, 2022.

[10] Yikai Wang, Xinghao Chen, Lele Cao, Wenbing Huang, Fuchun Sun, and Yunhe Wang. Multimodal token fusion for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12186–12195, June 2022.

[11] Nina Shvetsova, Brian Chen, Andrew Rouditchenko, Samuel Thomas, Brian Kingsbury, Rogerio Feris, David Harwath, James Glass, and Hilde Kuehne. Everything at once – multi-modal fusion transformer for video retrieval, 2022.

[12] Adrià Recasens, Jason Lin, João Carreira, Drew Jaegle, Luyu Wang, Jean baptiste Alayrac, Pauline Luc, Antoine Miech, Lucas Smaira, Ross Hemsley, and Andrew Zisserman. Zorro: the masked multimodal transformer, 2023.