Video-Text Retrieval

Index-based video retrieval
A study of video-language modeling for retrieval

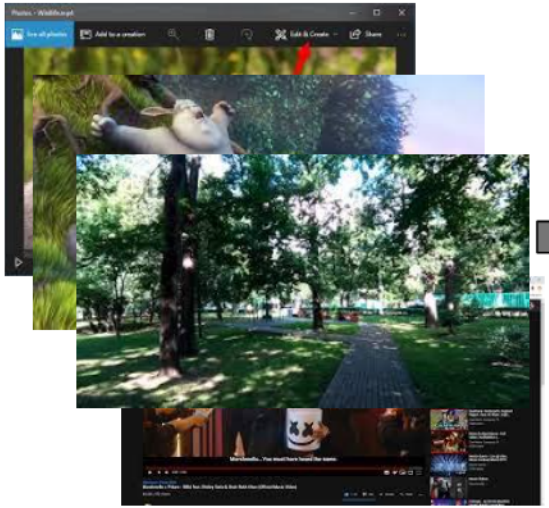
Sagar Joshi (2020701007)
Independent Study under Prof. Vineet Gandhi

Index-based retrieval of videos

Why index-based video retrieval?

- Current methods in video-text retrieval are directed towards developing embeddings to map videos and text to the same space.
- Consider a video library of 100M videos or so.
- Not only is embedding matching slower, many SoTA models need to have both - video and text for getting the embeddings, hence the process cannot be optimized with index as well.
- Index-based retrieval being much faster, can reduce the search space for retrieval of videos.

General pipeline for index-based retrieval



Video library

Video to text equivalent

Text query

Inverted index mapping text content to videos

Retrieved video set

What could be the possible ‘blackboxes’ for video-to-text approximation?

1. **Object detection** [images, videos]
   Getting all the objects in video frames to build an index. The performance here is limited to the **number of classes** an object detector is able to classify.
2. **Large-scale classification** [images]
   Inspired by CLIP, passing a large number of engineered prompts, and using the predictions of the model to get the top k prompts to build an index. Need to have such **well-engineered prompts**, however.
3. **Captioning** [images, videos]
   Using reverse of the task, i.e. converting video to textual description to build the index. The performance here depends on the **precision and coverage** of description given by the captioning model.

Initial Exploratory Study

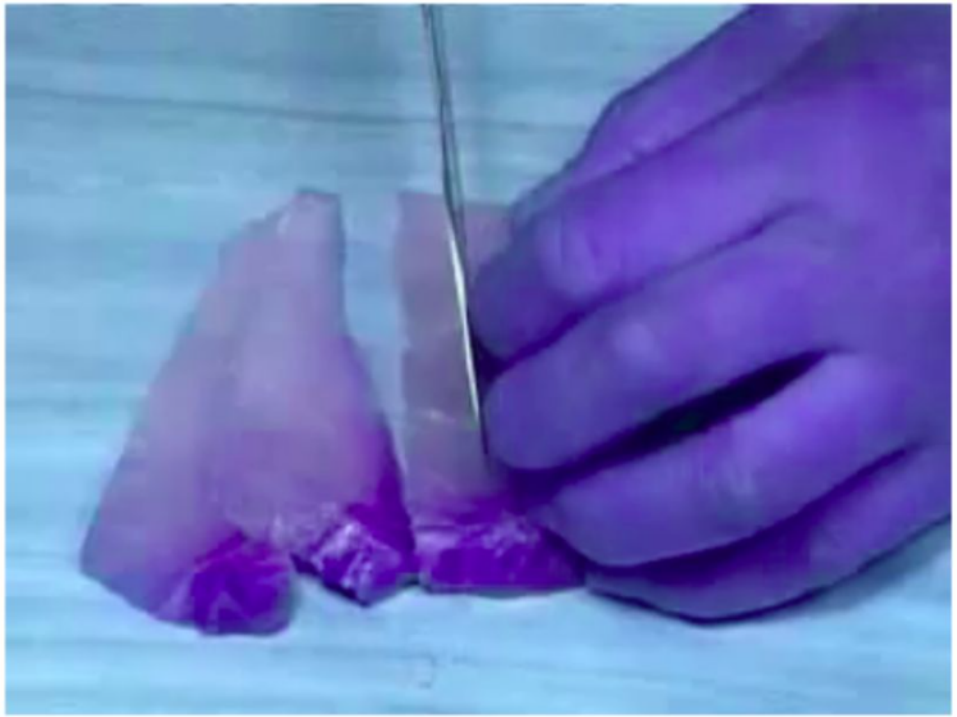
- An initial survey to check the feasibility of object detection and large scale classification for index-based video retrieval was done using the MSVD dataset.
- Both the approaches were image-based, by sampling one frame per second from the videos.
- No training / finetuning was performed on the models, since the idea was to see the ease of modeling using the respective approach.
- Focus was on the qualitative results given by using the models off-the-shelf on differing sampled frames from videos.

MSVD Dataset

Collecting Highly Parallel Data for Paraphrase Evaluation, ACL 2011

- Created as a resource for paraphrase evaluation by manually annotating YouTube videos of approximately 10s duration (on average)
- The dataset consists of 2089 video clips and 85,550 descriptions (average of 41 annotations per video) from 668 annotators
- The videos have a median FPS of 30, with 240 frames per video and a resolution of 360 x 488.
- Manual observations through the data showed the presence of many noisy, low quality descriptions for videos

MSVD Example

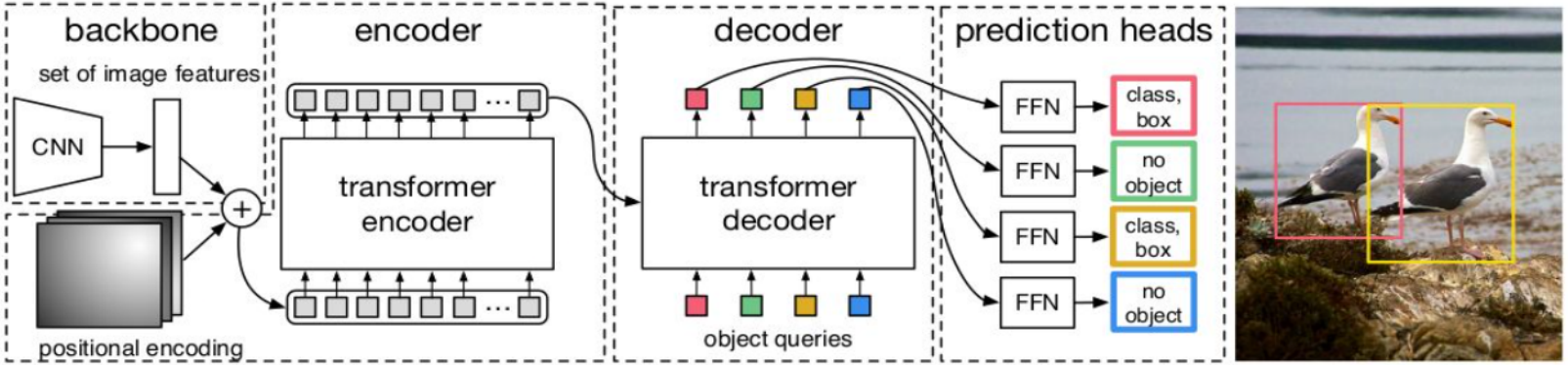


```
['a person is cutting a piece of fish into slices',
 'a piece of meat is being sliced',
 'a person is slicing fish',
 'a person is slicing a meat into pieces',
 'a person is cutting meat into strips with a knife',
 'a person is cutting up some meat',
 'men cutting fish slices with knief',
 'a man cuts up meat',
 'a woman is cutting fish',
 'a person is slicing up some meat',
 'a woman slices a fish',
 'the woman is cutting raw fish',
 'a person is cutting of meat',
 'someone slices raw meat',
 'someone is cutting a red piece of meat into thin strips',
 'a woman is slicing fish',
 'someone is cutting a piece of meat',
 'someone is slicing raw meat into strips',
 'the woman is slicing raw fish',
 'choping meat'])
```

A lot of annotations are based on assumptions taken from the video clip, hence have differences in individual descriptions.

DETR-based Object Detection

End-to-End Object Detection with Transformers, arXiv 2020

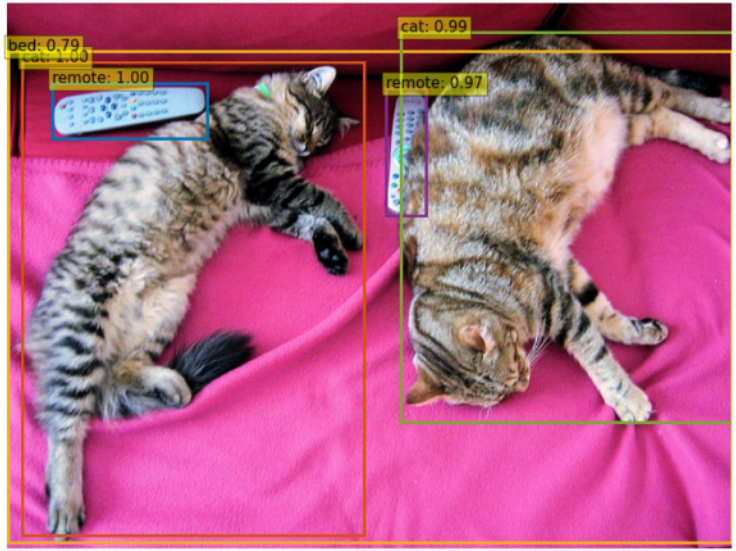


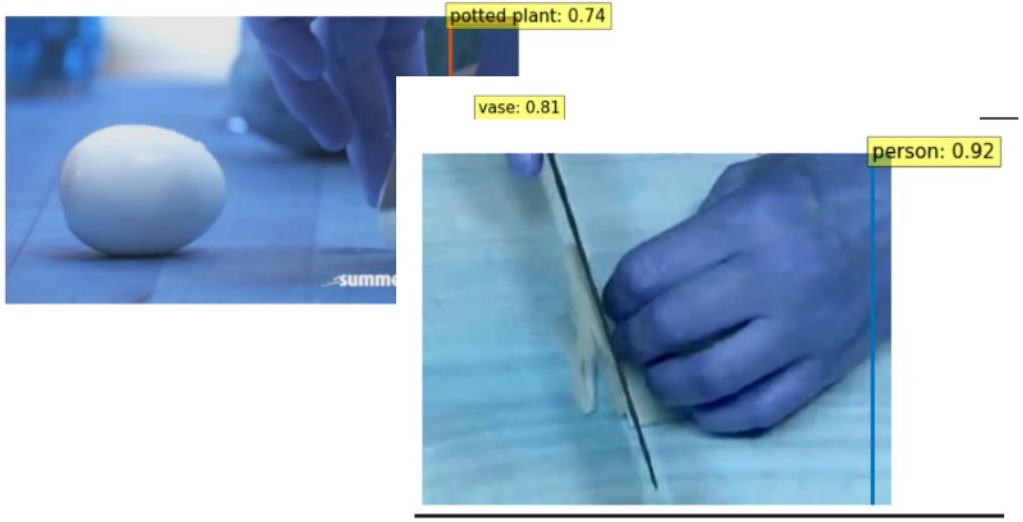
DETR uses a CNN backbone for image feature representation, followed by a complete transformer pipeline for object detection. CNN output is augmented with positional encoding to form object queries which are attended to by the transformer encoder & decoder. On the representation given by the decoder, the shared FFN parameters predict the bounding boxes and object tags.

DETR-based Object Detection

- DETR is trained for object detection on the COCO dataset, which consists of 80 classes of objects and training size of 118k images.
- It obtained competitive performance to Faster R-CNN (ResNet-50, ResNet-101) baselines on the task.
- However, the ability of detection is limited to these 80 broad classes - severely limiting the scope for finer retrieval based on queries.
- Importantly, DETR was found to perform well on images from the in-domain dataset, however, the performance was dismal on low-quality frames taken from the videos in MSVD dataset.

DETR-based Object Detection: Example

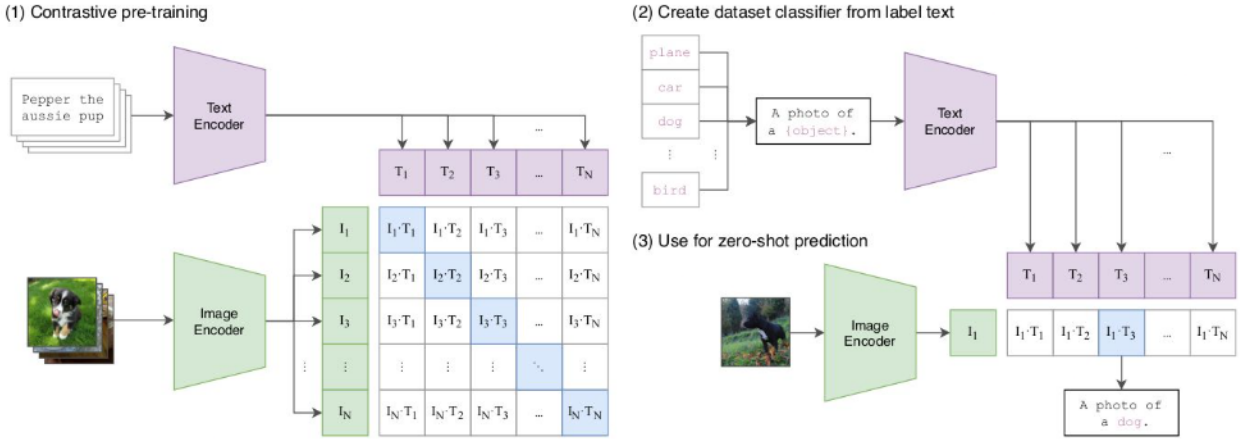




Stark difference in the performance on an in-domain, good quality image sample (left) v/s an out-of-domain, low quality image frame from a video (2 images on the right).

CLIP: Contrastive Language-Image Pretraining

Learning Transferable Visual Models From Natural Language Supervision, arXiv 2021



CLIP was learned by large scale contrastive pretraining to associate captions (natural language supervision) to images (visual concepts). It was trained on a large - 400M - unsupervised dataset consisting of image-text pairs found on the web “in the wild”, using contrastive learning to train the model to associate the right image-text pairing from a batch of N x N pairings (N being 32,768). The model displays superior zero-shot capabilities on various image classification datasets. High quality of image and text embeddings from CLIP have encouraged its adaptation to many diverse downstream tasks.

CLIP: Contrastive Language-Image Pretraining

- The image encoder in CLIP consists of a ResNet or a ViT architecture, while the text encoder is a transformer.
- A major advantage of the architecture lies in there being separate text and image encoder pipelines, avoiding extensive pairwise computation of every image and text embedding pair (as opposed to multimodal architectures like VisualBERT).
- Using CLIP in a zero-shot setting for image classification involves taking the softmax distribution over the cosine similarity of an image against given text prompts (or classes).
- Optimizing the performance needs some effort at prompt engineering and prompt ensembling.

```
# image_encoder - ResNet or Vision Transformer
# text_encoder  - CBOW or Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l]       - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t             - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T)  #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

Analyzing text prompt retrieval for MSVD clips

- Since the text descriptions in MSVD were observed to contain noisy, inaccurate samples, top 20 texts were selected per video by making use of CLIP, max pooling over the predictions for all the frames sampled at the rate of 1/s.
- A set of 40 videos were selected, each having 20 descriptions. For each video, the best 20 ranked descriptions were computed from the total of 800 text descriptions, which were matched against the actual top 20 texts for that video.
- Metrics from this **pilot experiment**:

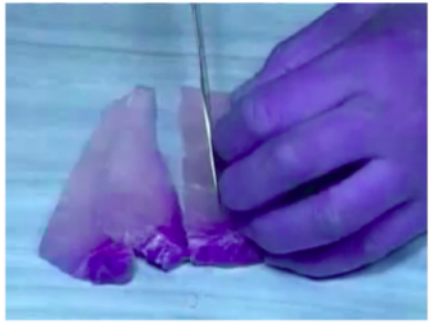
Top-1 Accuracy	Recall @ 10	BLEU-1	BLEU
15%	11.75%	43.43	19.76

- The performance on BLEU-1 score indicates decent amount of overlap between terms used in the actual prompts v/s retrieved ones. Other metrics, too, show encouraging performance considering the model is being tested for its zero-shot capabilities, and on images sampled from videos.

CLIP for text prompt retrieval: Example

The actual set of 20 prompts (bottom left) for a video (sampled frame in the top left), and the retrieved set of top 20 prompts (right) from the set of 800 prompts are shown alongside.

Although there is no exact match, the quality of selected prompts is very encouraging, and shows the understanding depth of the pretrained CLIP architecture.

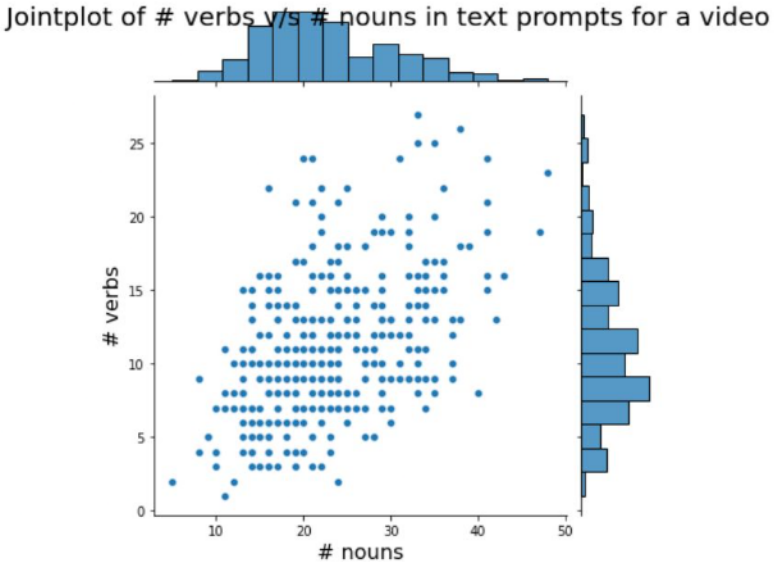


```
['a person is cutting a piece of fish into slices',
 'a piece of meat is being sliced',
 'a person is slicing fish',
 'a person is slicing a meat into pieces',
 'a person is cutting meat into strips with a knife',
 'a person is cutting up some meat',
 'men cutting fish slices with knief',
 'a man cuts up meat',
 'a woman is cutting fish',
 'a person is slicing up some meat',
 'a woman slices a fish',
 'the woman is cutting raw fish',
 'a person is cutting of meat',
 'someone slices raw meat',
 'someone is cutting a red piece of meat into thin strips',
 'a woman is slicing fish',
 'someone is cutting a piece of meat',
 'someone is slicing raw meat into strips',
 'the woman is slicing raw fish',
 'choping meat'])
```

```
['a person trims fat from a piece of meat',
 'a person is removing the fat and skin from a piece of chicken',
 'a person is trimming fat from some chicken',
 'a chef removes the skin and fat from chicken flesh',
 'a woman is removing fat from a piece of chicken',
 'a person cutting up raw chicken',
 'a person cuts the fat off chicken',
 'someone is trimming fat and skin off of pieces of chicken',
 'a woman is cutting the fat from a piece of meat',
 'a woman is trimming fat from chicken breast with kitchen shears',
 'someone is using scissors to cut fat from chicken',
 'a lady is cutting the skin of the chicken',
 'a woman takes the excess fat off of a piece of meat',
 'a woman is trimming fat from poultry',
 'a woman is removing skin from some meat',
 'a person trims the fat off a piece of meat',
 'someone is cutting the skin and pieces of fat off of a chicken',
 'someone is removing skin and fat from meat',
 'a person is cutting fat from a piece of meat',
 'a woman is trimming the fat from meat'])
```

Using CLIP, Object Recognition Style

- Considering the large-scale classification capability of CLIP, two separate object vocabularies were constructed for nouns (2530) and verbs (981).
- The idea was to use the output of noun & verb recognition from videos to build an inverted index, thus matching queries to retrieve videos.
- This study was performed over 400 videos and 15,842 texts from MSVD.
- Top 20 nouns and verbs were computed per video, using the vocabulary as ‘classes’ for classification.



CLIP Noun & Verb Identification Examples

```
[('squirel', 0.1263),
 ('squirrrel', 0.0803),
 ('squirrel', 0.06976),
 ('chipmunk', 0.04434),
 ('rat', 0.04166),
 ('hamster', 0.02864),
 ('squerel', 0.01907),
 ('squirre', 0.01736),
 ('pet', 0.013954),
 ('peanut', 0.01271),
 ('video', 0.01139),
 ('singing', 0.01021),
 ('bird', 0.00795),
 ('patato', 0.00747),
 ('pepper', 0.006096),
 ('cleaning', 0.006),
 ('feeding', 0.005215),
 ('sparrow', 0.005135),
 ('hai', 0.005054),
 ('mammal', 0.004974)]
```

```
[('hop', 0.06024),
 ('pet', 0.0584),
 ('video', 0.04767),
 ('thump', 0.0371),
 ('slicimg', 0.02359),
 ('handfeeding', 0.01753),
 ('dash', 0.01726),
 ('ham', 0.01726),
 ('squirm', 0.01672),
 ('squirt', 0.01344),
 ('scamper', 0.01344),
 ('flap', 0.012054),
 ('sit', 0.00882),
 ('sweep', 0.008415),
 ('paw', 0.00731),
 ('sing', 0.007195),
 ('waddle', 0.007195),
 ('kangaroo', 0.007084),
 ('chew', 0.006973),
 ('spike', 0.006973)]
```

Video 1 Nouns Verbs

```
[('swordfight', 0.0774),
 ('contestant', 0.02553),
 ('parkour', 0.02397),
 ('sludgehammer', 0.01324),
 ('run', 0.01132),
 ('magician', 0.010315),
 ('freerunning', 0.010155),
 ('nunchuck', 0.00969),
 ('wand', 0.00954),
 ('pedal', 0.007545),
 ('trick', 0.0072),
 ('shaper', 0.006977),
 ('jumped', 0.00687),
 ('someonehand', 0.006454),
 ('hammer', 0.00635),
 ('jumping', 0.004795),
 ('kick', 0.004795),
 ('medic', 0.004574),
 ('leap', 0.0043),
 ('nun', 0.00423)]
```

```
[('parkour', 0.05063),
 ('kungfu', 0.04007),
 ('run', 0.02393),
 ('freerunning', 0.02144),
 ('count', 0.01805),
 ('escape', 0.01569),
 ('trick', 0.015205),
 ('backflips', 0.013626),
 ('hammer', 0.01342),
 ('saw', 0.011475),
 ('devein', 0.01112),
 ('dab', 0.01078),
 ('kick', 0.01013),
 ('leap', 0.00908),
 ('jumproping', 0.00826),
 ('hook', 0.00789),
 ('jump', 0.007526),
 ('fell', 0.007526),
 ('topple', 0.007183),
 ('tap', 0.007183)]
```

Video 2 Nouns Verbs

Analysis of CLIP for possibility of index creation

- There was some degree of relevance in the identified nouns and verbs, however, it didn't seem to be significant enough to build an inverted index for video retrieval.
- Some effort put in effectively designing prompts for retrieving the nouns and verbs should better the performance, since CLIP has been trained to classify using natural sentences after all.
- Nevertheless, the performance of CLIP is very encouraging considering it was used without any finetuning on the dataset, and any use of CLIP embeddings for further finetuning on the task (classical video-text retrieval) is expected to give exemplary performance.

Video Captioning for Video-Text Retrieval

- Video captioning is a natural approach for getting suitable text descriptions from videos for the purpose of indexing, since the models for video captioning have been trained for the same task.
- Stronger performance on video captioning is thus expected to give a better quality index to be used for video retrieval.
- Subsequent study was thus carried out using video captioning using the UniVL architecture on the MSR-VTT dataset - a larger, better quality dataset commonly used for benchmarking on video captioning as well as retrieval.

About the MSR-VTT dataset

"MSR-VTT: A Large Video Description Dataset for Bridging Video and Language", CVPR 2016

- The dataset comprised of 10K web video clips (41.2 hrs) and 200k clip-sentence pairs, with each video having approx. 20 sentences, manually annotated by AMT workers.
- Superiority wrt the other datasets before it (MSVD, YouCook, M-VAD, TACoS, MPII-MD) is in the diversity of video content (not just cooking/movies) and the size of annotated video data.
- Initial experiments were started off using MSVD dataset, but later shifted to MSR-VTT due to its size and ubiquity throughout the literature.

MSR-VTT Examples

1. A black and white horse runs around.
2. A horse galloping through an open field.
3. A horse is running around in green lush grass.
4. There is a horse running on the grassland.
5. A horse is riding in the grass.

1. A woman giving speech on news channel.
2. Hillary Clinton gives a speech.
3. Hillary Clinton is making a speech at the conference of mayors.
4. A woman is giving a speech on stage.
5. A lady speak some news on TV.

1. A child is cooking in the kitchen.
2. A girl is putting her finger into a plastic cup containing an egg.
3. Children boil water and get egg whites ready.
4. People make food in a kitchen.
5. A group of people are making food in a kitchen.

1. A man and a woman performing a musical.
2. A teenage couple perform in an amateur musical
3. Dancers are playing a routine.
4. People are dancing in a musical.
5. Some people are acting and singing for performance.

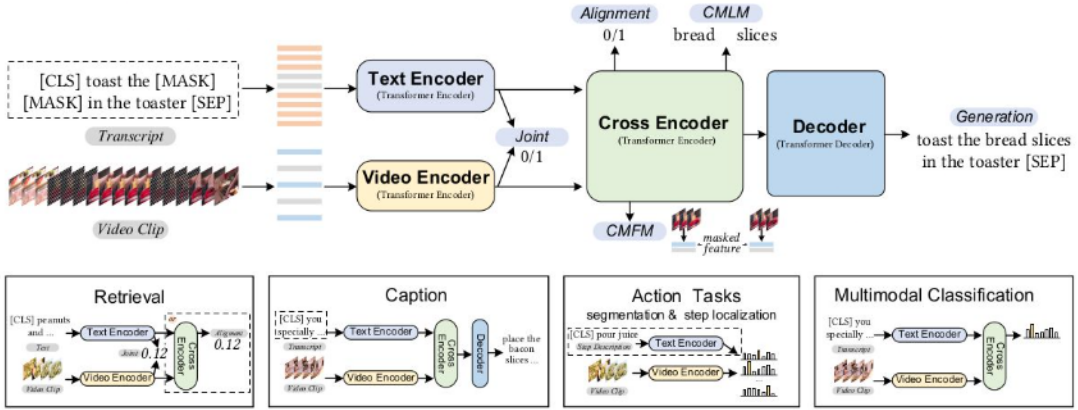
1. A white car is drifting.
2. Cars racing on a road surrounded by lots of people.
3. Cars are racing down a narrow road.
4. A race car races along a track.
5. A car is drifting in a fast speed.

1. A player is putting the basketball into the post from distance.
2. The player makes a three-pointer.
3. People are playing basketball.
4. A 3 point shot by someone in a basketball race.
5. A basketball team is playing in front of speculators.

The descriptions for a video may
- focus on **differing actions** (cooking v/s put finger in a plastic cup) in the video
- choose to deal with only a **select entities** (player v/s basketball team)
- can describe in differing levels of **specificity** (woman v/s Hillary Clinton, singing and acting v/s performing a musical).

UniVL

UniVL: A Unified Video and Language Pre-Training Model for Multimodal Understanding and Generation, arXiv 2020



The UniVL architecture consists of a text encoder (BERT), video encoder (Transformer layer) jointly connected to a cross-modal encoder (transformer), followed by a transformer decoder. The decoder helps in pretraining to add generation capabilities by learning to reconstruct the input text, while benefits downstream generation tasks like captioning by having the ability to finetune for the text labels.

UniVL: Pretraining Objectives

1. Video-Text Joint: Retrieval-oriented operation to align the unimodal text & video encoder representations of transcript to a video clip, trained using the popular MIL-NCE objective
2. Conditional Masked Language Model: Reproduce 15% of the masked tokens in text
3. Conditional Masked Frame Model: NCE objective to identify the correct frame against negative distractors for 15% of the masked frames (masked vectors) in the video clip
4. Video-Text Alignment: Predict scores on the crossmodal video-text representation, learned using NCE objective
5. Language Reconstruction: Autoregressive decoder training to reconstruct input text, endowing generation capability for downstream tasks

More on UniVL Pretraining

- UniVL was pretrained on 1.2 M videos and ASR transcripts from Howto100M dataset, each video on average being 6.5 min long, and having 110 clip-text pairs.
- Two pretraining strategies:
  - StagedP - Only the individual text & video encoders were learnt using the joint objective, following which rest of the objectives were introduced on a decreased learning rate
  - EnhancedV - 15% of the clip-text tokens had the entire text portion masked out to have a better learnt video representation
- Text embeddings are initialized from BERT-base, while for videos, 3D features are extracted from the S3D model pretrained using MIL-NCE objective on Howto100M dataset

UniVL: Finetuning tasks & datasets

UniVL had SoTA results on finetuning for 5 different tasks across these datasets:

1. Text-based video retrieval (Youcook2, MSR-VTT)
2. Video captioning (Youcook2)
3. Action segmentation (COIN)
4. Action step localization (CrossTask)
5. Multimodal sentiment analysis (CMU-MOSI)

Training UniVL for Caption Generation

Pretrained weights provided by the authors were finetuned for caption generation on the train split of the MSR-VTT dataset, by using one textual description per clip.

The split consisted of 6513 train samples, 497 validation samples & 2990 samples in the test split.

The model was trained for 5 epochs on batch size of 32. Best checkpoint based on the performance on dev split was obtained after the second epoch, which was used to measure the performance on the test split; shown below -

BLEU-1	BLEU-2	BLEU-3	BLEU	METEOR	ROUGE-L	CIDEr
80.88	66.71	53.36	41.45	28.75	60.61	47.96

Using UniVL-generated captions for indexing

- Captions generated on the test split of MSR-VTT by the finetuned UniVL model were used to build an inverted index of tokens to be used for video retrieval from text queries.
- The captions were decoded using Beam Search decoding, using a beams size of 5.
- A BLEU score of 41.45 measured against all the references in the dataset indicates a decent overlap of the generated caption with the content from the references.
- For retrieving using the constructed index, one description was chosen per text video for each of the 2990 videos in the test split.

Index construction

The index was constructed using Whoosh, a pure Python-based search engine using the Okapi BM25 score for ranking

$$\text{score}(D,Q) = \sum_{i=1}^{n} \text{IDF}(q_i) \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D) + k_1 \cdot \left(1 - b + b \cdot \frac{|D|}{\text{avgdl}}\right)} \qquad \text{IDF}(q_i) = \ln\left(\frac{N - n(q_i) + 0.5}{n(q_i) + 0.5} + 1\right)$$

D = document for which the score is computed against the query
Q = query containing keywords q_i
f(q_i, D) = term frequency of q_i in D
|D| = length of the document in words
avgdl = average length of a document in the text collection dealt with
N = total # documents
n(q_i) = # documents containing q_i
k_1, b = free parameters

More on Indexing

- Apart from stemming, no other text preprocessing was done on the queries and indexed captions.
- The retrieved list of documents was examined for recall at k (R@k) and mean/median rank values.
- Since we're dealing with an approximated index, the values of k for R@k were taken over a large range.
- The original text descriptions were indexed themselves to verify the sanity of the index construction & retrieval.

Recall									Rank	
1	5	10	20	50	100	200	500	1000	Mean	Median
95.35	99.33	99.90	100	100	100	100	100	100	1.11	1.0

Indexing Results

The results obtained on the index created using UniVL-generated captions:

Recall									Rank	
1	5	10	20	50	100	200	500	1000	Mean	Median
2.07	7.46	11.37	17.39	25.18	32.44	41.14	51.74	64.62	836.81	427

It should be noted here that the scores are not indicative of the good BLEU score (41.45) obtained for caption generation, since it was measured against all the 20 references in the dataset. When BLEU was measured using only one reference per caption, it turned out to be a mere 4.59.

Adding more content for higher recall: survey experiment

Since the BLEU score computed with individual reference text was drastically low as compared to the one computed using all references, more content was needed in querying.

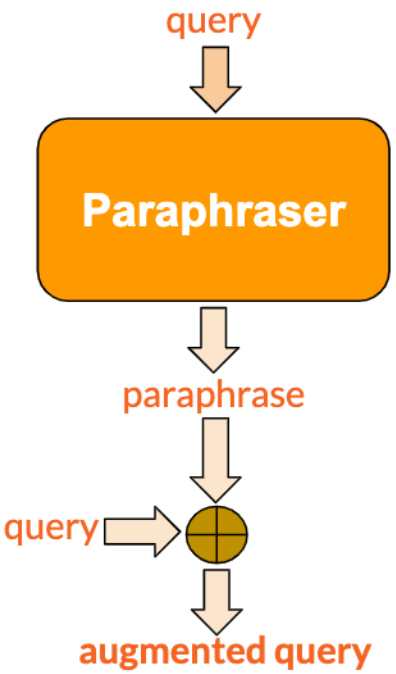
The variation in recall and rank scores was observed by adding increasing number of direct descriptions from the data to the caption as the augmented query.

An improvement in values indicated that a mechanism to let the model hallucinate more related content than the generated caption would benefit the retrieval.

Results on Adding Original Descriptions (Captions) to Query

# captions added	Recall									Rank	
	1	5	10	20	50	100	200	500	1000	Mean	Median
0	2.07	7.46	11.37	17.39	25.18	32.44	41.14	51.74	64.62	836.81	427
1	2.37	7.99	11.70	17.59	26.52	33.81	41.87	55.75	68.16	777.74	349
2	3.04	10.77	15.82	23.21	34.38	42.91	53.21	65.99	76.69	587.77	159
5	3.48	11.57	17.79	25.18	37.46	49.40	61.84	75.65	85.62	404.37	104
10	3.54	12.27	18.93	27.49	40.37	52.37	65.65	79.53	89.26	332.74	89

Paraphrasing for query expansion

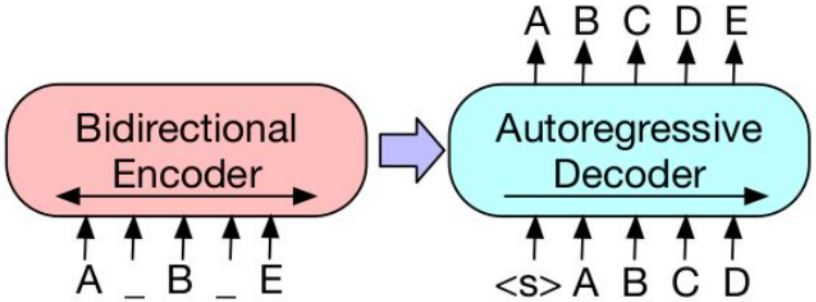


A paraphraser can help in augmenting the query with more terms.

The added text may not be perfectly relevant, the goal here is to only increase the recall.

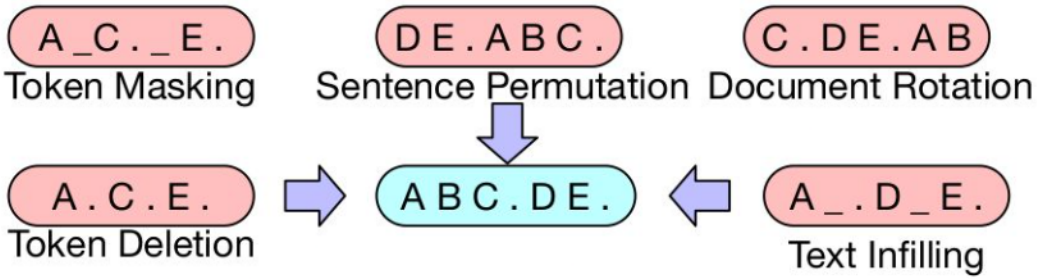
BART architecture was trained on the train split of MSR-VTT dataset for generating paraphrases.

BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, ACL 2020



BART, termed as a denoising autoencoder, architecturally consists of a bidirectional encoder (BERT-like) and an autoregressive decoder (GPT-like) - essentially a seq2seq transformer architecture with a few modifications. The architecture pretrained using 5 denoising objectives achieved competitive performance on discriminative tasks and SoTA performance on generative tasks on finetuning.

BART: Denoising objectives



The input is corrupted using these noise transformations, and the model is trained to correctly reproduce the original input.

1. Token masking: BERT-like MLM, replacing random tokens with [MASK]
2. Token deletion: Tokens are deleted from the input, the model should decide where they were
3. Text infilling: Spans of differing lengths are replaced with [MASK]
4. Sentence permutation: Sentences (divided by full stops) are randomly permuted
5. Document rotation: Choosing a random, the document (input) is rotated to start with that token

Training BART for paraphrase generation

- From the 20 text descriptions available per video, 10 pairs were to act as paraphrase input-outputs.
- The model was trained for 2 epochs with a batch size of 16 (across 2 GPUs), learning rate having a warm up till 0.3 (max value 5e-5) of the steps followed by a linear decay, AdamW optimization.
- Predictions were initially taken using beam search decoding with a beam size of 5, however, the paraphrases were not observed to significantly differ from the input.
- The primary goal was to have the same concept expressed using differing terms, or adding some contextual information to the same, hence more diversity was expected.

Decoding for the most diverse paraphrase

In order to get a better, diverse paraphrase from the trained model, the following strategy was adopted:

- Nucleus sampling with a p value of 0.6 and temperature set to 0.8 was used to decode outputs
- 5 sequences were sampled per input query
- The best paraphrase was chosen as the one that had the least lexical (unigram) overlap with the input

Metrics for Paraphrase Evaluation

The improvement in lexical diversity can be seen from the Self-BLEU (lower the better) and PINC (higher the better) metrics for paraphrase evaluation.

	Self-BLEU	PINC
Beam search decoding	44.79	45.52
Nucleus sampling	32.02	58.71

Examples of improved lexical diversity in paraphrases

Input	Beam search decoding	Nucleus sampling followed by best paraphrase selection
a group of people kids dancing and playing music	a group of people dancing and playing music	people are dancing and singing
people are doing karate	people are doing karate	two men are fighting
a person is using a makeshift stove to melt chocolate	a person is using a stove to melt chocolate	a person is making some food
a man describes how to skin a snake	a man shows how to skin a snake	a man is cutting a snake
an old man and woman is hit by an arrow while seated outside of a restaurant	an old man and woman are hit by an arrow	a man and woman are talking

Improvement in Indexing with Paraphrase Augmentation

method	Recall									Rank	
	1	5	10	20	50	100	200	500	1000	Mean	Median
original	2.07	7.46	11.37	17.39	25.18	32.44	41.14	51.74	64.62	836.81	427
with paraphrase (beam)	2.04	8.19	12.78	19.33	28.03	36.12	46.69	58.83	71.10	713.56	249
with paraphrase (nucleus)	2.17	8.16	12.98	19.73	28.73	37.09	48.80	61.57	73.71	660.68	212.5
with actual caption	2.37	7.99	11.70	17.59	26.52	33.81	41.87	55.75	68.16	777.74	349

Analysis over the results

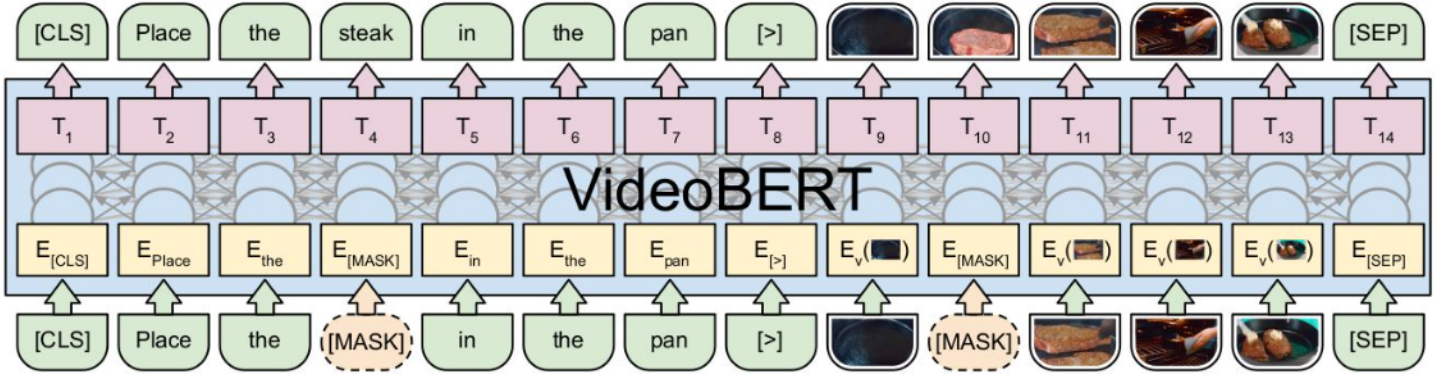
1. Augmenting with a paraphrase turned out to be successful, with the scores for the same exceeding the results of even the experiment using one actual caption.
2. The better diversity of the nucleus-sampling based decoding process was shown in the retrieval results as well.
3. A large size of retrieved results (~1000) needs to be considered for having a decent chance of the actual video being retrieved.
4. The study being conducted on the test split of MSR-VTT with 2990 samples, it is to be known how much would the recall/rank values suffer for much larger datasets (100M or so) for which it is intended.

A study of video-language modeling for video retrieval

Video-Language Multimodal Architectures

- The SoTA in video-text retrieval has been achieved by multimodal transformer architectures dealing with video and text modalities.
- All of these models are trained to build the representations for a given pair of video and text such that they are aligned in a common space.
- Embedding representations, though involving much larger compute & time, are naturally superior than simple text-based matching because of their ability to represent a lot of information with a point in n-dimensional space.
- A survey of some of the SoTA multimodal architectures was carried out. The performance of some of them on the video retrieval was computed by training on the MSR-VTT dataset.

VideoBERT: A Joint Model for Video and Language Representation Learning, ICCV 2019

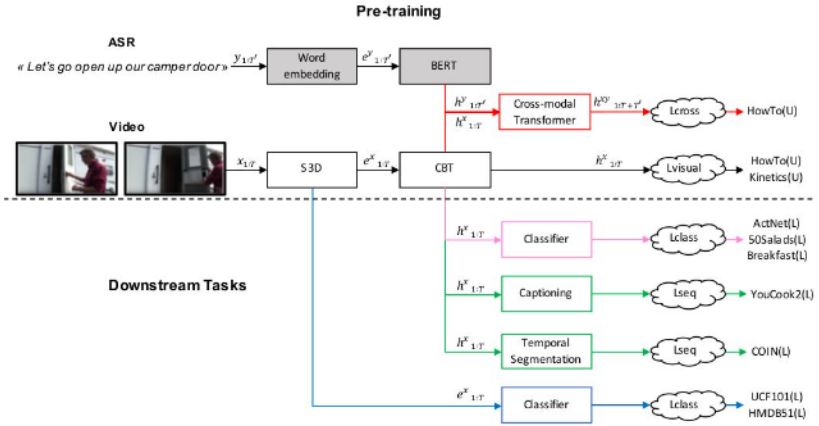


In this first transformer-based modeling of video and text, video was processed as a series of quantized ‘visual tokens’ while basing the architecture on BERT. The model was trained on a collected dataset of 312k videos by self-supervised training using MLM objective (only text, only video, video & text) on sequence of text (ASR stream) and visual tokens. The model was finetuned on the supervised YouCook II dataset for video captioning and checking zero-shot performance on noun and action prediction based on video.

More on VideoBERT

- The videos were processed by samping at 20 fps followed by average pooling S3D (pretrined on Kinetics dataset) representation of every 30-frame clip.
- Hierarchical k-means was used to cluster the video features into 4 hierarchy levels and 12 clusters per level, yielding 20,736 (12^4) clusters.
- ASR was broken into sentences by adding punctuations using an off-the-shelf LSTM, and video segments were broken per sentence.
- Pretrained checkpoint of BERT-large was used to initialize, following which VideoBERT was pretrained for 0.5M iterations (~8 epochs).
- In zero-shot noun and verb prediction, a fixed templated sentence “now let me show you how to [MASK] the [MASK]” was used to extract the noun and verb predictions. Top-1 & top-5 classification metrics were calculated form it.
- Impressive qualitative performance was also shown on text-to-video prediction and video-to-video prediction.

Learning Video Representations Using Contrastive Bidirectional Transformer, arXiv 2019



The CBT architecture consists of a BERT model for text representation, CBT model for video representation and cross-modal CBT for combined representation. The BERT model is frozen in pretraining. NCE is used to train the CBT model for correct frame identification the cross-modal CBT for maximizing the mutual information between video and text streams. Kinetics and Howto100M datasets were used for pretraining this model.

HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips, ICCV 2019

- A dataset consisting 136 million video clips from 1.22M narrated instructional videos describing over 23.6k visual tasks was introduced for large-scale video-text pretraining.
- 12 categories of tasks from WikiHow were used to first query YouTube videos for the task, followed by filtering to select appropriate videos.
- Joint representation for video & text was learnt by optimizing for max-margin ranking loss, negative samples being half intra-video & half from other random videos.
- In modeling, concatenation of 2D and 3D CNN outputs was done for videos, while 1D CNN was applied over word2vec embeddings for text: impressive zero shot, finetuning achieving SoTA

Experiments on the HowTo100M model

Zero-shot performance of the pretrained weights on MSRVTT for text to video retrieval:

Recall			Rank
1	5	10	Median
7.50	21.20	29.60	38.0

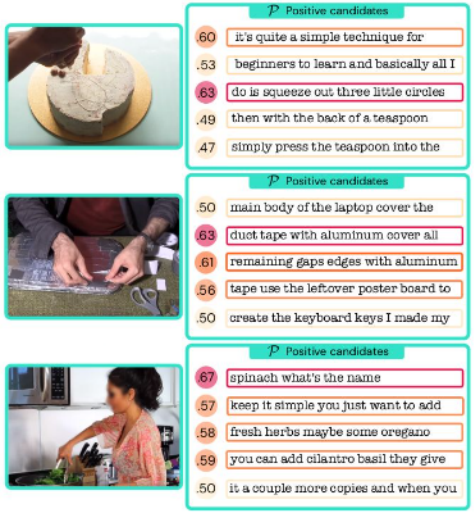
The model was also finetuned for 50 epochs using pre-extracted features provided by the authors. Training hyperparameters were kept the same as provided in the script. Results on the best performing checkpoint:

Recall			Rank
1	5	10	Median
17.40	41.00	55.60	8.0

Misalignments in HowTo100M & MIL

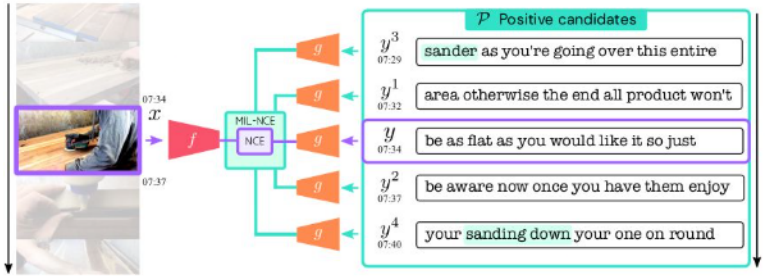
End-to-End Learning of Visual Representations from Uncurated Instructional Videos, CVPR 2020

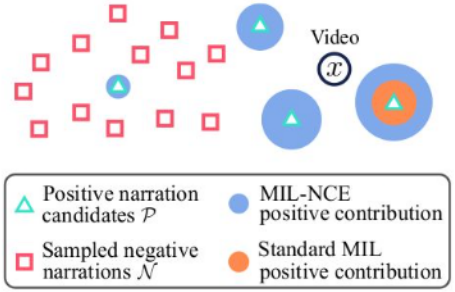
- Since the dataset is not manually annotated, the semantically corresponding text for a clip may be within a window around the clip, before or after the temporal pairing in video.
- Around 50% of the instances in HowTo100M are not aligned.
- Multiple Instance Learning (MIL) allows having multiple positive candidate pairs for contrasting against negative samples.
- The joint probability of a set of positive candidate pairs is maximized.



MIL-NCE

- The MIL-NCE objective, bespoke for noisily aligned data, maximizes the ratio of the sum of positive candidate scores to that of negative candidate scores.
- The parameterized mappings f for video and g for text are thus trained to map to a joint embedding space.
- For NCE, the negatives are sampled symmetrically - negatives for both, video and text are taken.
- A 3D CNN-based video encoder and word2vec-based text encoder achieved SoTA on text-to-video retrieval and action-based tasks.





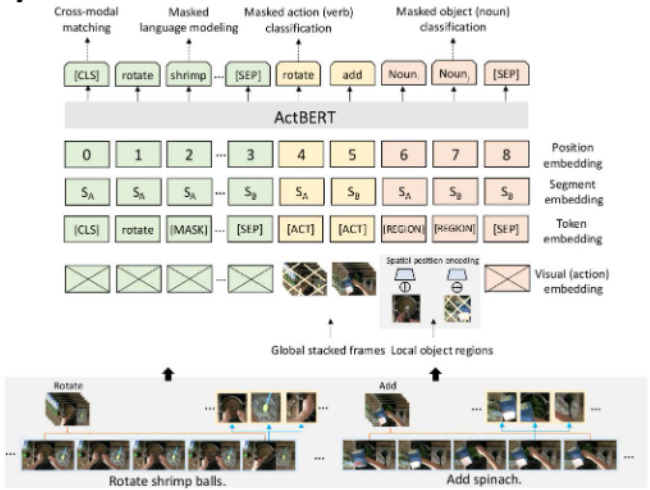
$$\max_{f,g} \sum_{i=1}^{n} \log \left(\frac{\sum_{(x,y)\in\mathcal{P}_i} e^{f(x)^\top g(y)}}{\sum_{(x,y)\in\mathcal{P}_i} e^{f(x)^\top g(y)} + \sum_{(x',y')\sim\mathcal{N}_i} e^{f(x')^\top g(y')}} \right)$$

Training UniVL for Video Retrieval

The UniVL proposed a joint video-text encoder for modeling the cross-modality. Pretrained weights of the model provided by the authors were finetuned for 4 epochs (batch size 8, learning rate 5e-5) on MSR-VTT for video retrieval.

Recall			Rank
1	5	10	Median
22.50	51.40	66.30	5.0

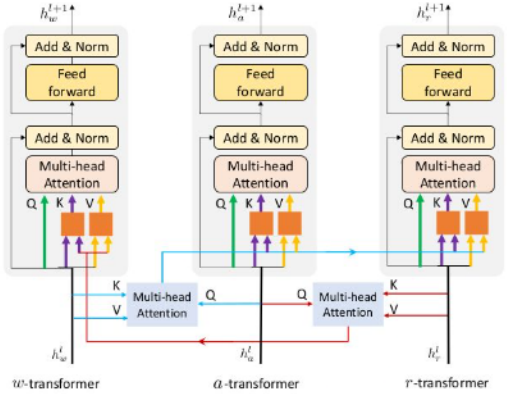
ActBERT: Learning Global-Local Video-Text Representations, CPVR 2020



ActBERT architecture combines the action, object and linguistic features in a video using its representation of ‘Tangled Transformer’ having co-attentional blocks between the three inputs. Special embeddings were used for adding information on different aspects of the input. The pretrained model outperformed its non-transformer baselines on 5 downstream tasks.

ActBERT: Tangled Transformer

- The tangled transformer consists of three separate streams for encoding the action features (a-transformer), linguistic features (w-transformer), and object features (r-transformer).
- a-transformer catalyzes mutual interactions by querying in the w-transformer and r-transformer.
- The generated blended features by co-attention from r-transformer are fed to w-transformer, and those from w-transformer are fed a-transformer and r-transformer.
- Key-value pairs are generated within each transformer using the blended features as input.



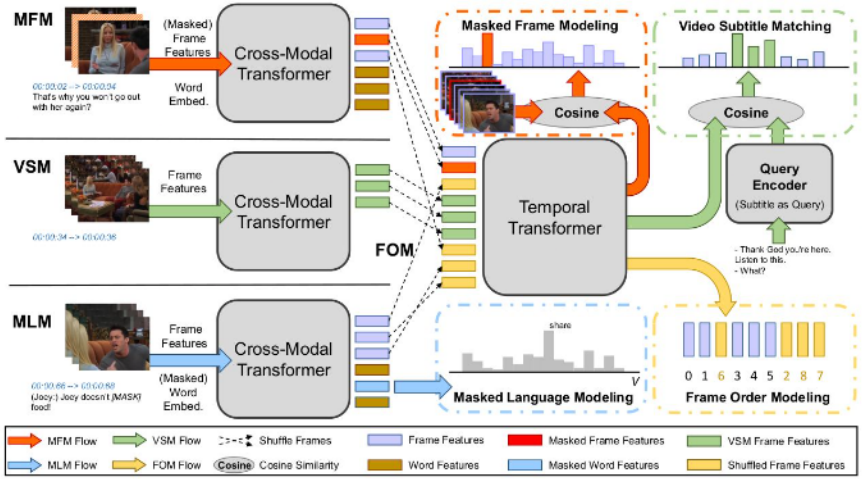
$$c_w = Multihead(W_q^1 h_a^l, W_k^w h_w^l, W_v^w h_w^l),$$
$$c_r = Multihead(W_q^2 h_a^l, W_k^r h_r^l, W_v^r h_r^l),$$

ActBERT: Embeddings & Pretraining

Input consists of a sum of 4 embeddings:
Positional embedding - Learnable positional embedding indicative of temporal order. It remains the same for regions extracted from the same frame.
Segment embedding - Same across a clip. Used to distinguish multiple clips for long-term video context modeling.
Token embedding - WordPiece embeddings for linguistic tokens and special tokens to represent action & region features
Visual (action) embedding - A 3D convolutional network is trained to identify verbs in a video clip (4s, 32 frames). Global average pooled features from the trained model are used.

The model is trained on 4 self-supervised tasks of Masked language modeling (predict over masked tokens), Masked action classification (predict over verb vocabulary for masked action features), Masked object classification (predict over object vocabulary for masked image region) and Cross-modal matching (relevance of sentence to video, NSP-like) on HowTo100M dataset.

HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training, EMNLP 2020



HERO consists of a transformer-based hierarchical architecture: the first stage - cross-modal transformer - for fusing video & text modalities & the second stage - temporal transformer building a sequence-aware, global context. The model was pretrained using 4 objectives on HowTo100M and TV (more challenging) datasets.

HERO: Pretraining Tasks

- **Masked Language Modeling (MLM):** Predict 15% masked tokens from the subtitle replaced by [MASK] making use of unmasked tokens and visual frames
- **Masked Frame Modeling (MFM):** Predict 15% masked frames making use of unmasked frames and the subtitle. Two variants experimented:
  - Masked Frame Feature Regression - An FC layer converts the output frame representation to a vector in the same space as the input frame embedding. L2 regression is applied between the output of the masked frame passed through this FC and the input frame embedding.
  - Masked Frame Modeling with NCE - Using the output vector computed as in regression, softmax NCE is used to identify the input frame versus negative distractors. This loss was chosen in the final design.

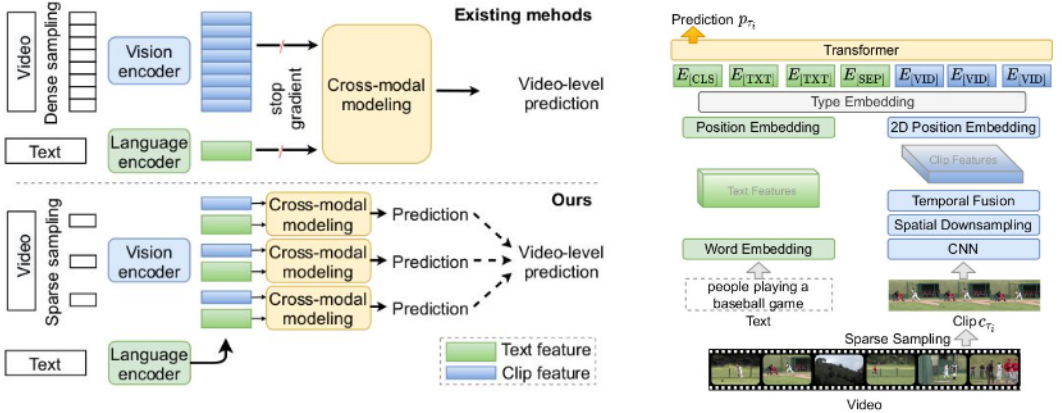
HERO: Pretraining Tasks (contd.)

- **Video-Subtitle Matching (VSM):** Calculated between the output of the temporal transformer for the video frames and text query representation from the cross-model transformer
  - Local alignment - Identify the start and end points in the video from the query
  - Global alignment - Max-pooled cosine similarity between the query and every frame is taken as the query-video score. Hinge loss between is computed to maximize the score for positive samples by using 2 negative samples: representing a wrong query, representing a wrong video.
- **Frame Order Modeling (FOM):** Frames are reordered after the output from the cross-modal transformer, the output from temporal transformer is trained to predict the right temporal order for each frame.

More on HERO

- Video embedding is prepared by concatenating the representations of 2D ResNet pretrained on ImageNet and 3D SlowFast pretrained on Kinetics, passed through an FC.
- Text embeddings are calculated by adding wordpiece token embeddings with positional information.
- Residual connection is added from the embeddings to the input of the temporal transformer to avoid losing positional information.
- When required, only-text representation (like query in VSM) is computed by passing text to the cross-modal encoder, with video input being zero.
- Two new benchmarks were introduced for video-language modalities:
    - **How2R** - Annotated on 9,421 videos on selected segments (10-20s) from a larger scene for text-based video-moment retrieval. Dataset size is 51,390 queries
    - **How2QA** - Human written question-answers having one right answer out of 4 choices per answer for video question-answering. Dataset size is approx 44k QA pairs for 22k 60s clips from 9k videos.

Less is More: CLIPBERT for Video-and-Language Learning via Sparse Sampling, CVPR 2021



CLIPBert uses a 2D CNN image encoder, BERT-based text encoder and a cross model transformer for fusing the representations of the two modalities. The key idea of CLIPBert is to avoid offline pre-extracted features by sparsely sampling features from short video clips, thus working directly at pixel level. The model exhibits the capability to work on full-range videos, benefiting from representations from sampled short clips.

More on CLIPBERT

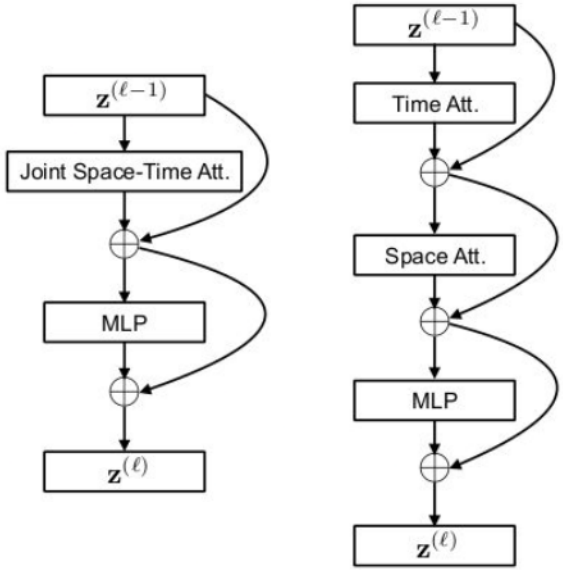
- One or few sampled clips are used at training, however, predictions over densely sampled clips are aggregated at inference time
- The model benefits from pretrained image-text models by using weights trained on COCO & Visual Genome captioning tasks in the 2D CNN-based image encoder. The model is pretrained on the same datasets.
- The sparse sampling strategy also serves as a data augmentation trick, and is shown to perform better than dense uniform sampling
- The model was finetuned on text-to-video retrieval and video question answering tasks.

The model architecture is, however, not particularly suited for text-to-video retrieval since it will involve pairwise computation between every possible text and video pair, and the embeddings cannot be precomputed for indexing.

Space-Time Attention

Is Space-Time Attention All You Need for Video Understanding?, ICML 2021

- This work proposed TimeSFormer, the first convolution-free approach for video representation, using only transformer-style self-attention
- Introduced ‘divided space-time attention’ effective for spatiotemporal data like video
- Inspired from ViT, each video frame is spatially divided into N patches each of size P^2
- The patches are encoded with learnable positional embedding to have z, which encodes the spatiotemporal information of the patch
- Query/key/value representation is computed for each patch from z



Joint Space-Time Attention (ST)

Divided Space-Time Attention (T+S)

Divided Space-Time Attention

A full self-attention over all T frames and P patches will take quadratic computation. In divided space-time attention, temporal attention is first computed by computing the softmax over query-key comparisons for a patch (p, t) with all the patches at the same spatial location, but all temporal locations.

$$\boldsymbol{\alpha}_{(p,t)}^{(\ell,a)\text{time}} = \text{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)\top}}{\sqrt{D_h}} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p,t')}^{(\ell,a)}\right\}_{t'=1,\ldots,F}\right]\right)$$

A weighted sum of the attention scores over the value vectors then gives the representation from a particular attention head after temporal representation.

$$\mathbf{s}_{(p,t)}^{(\ell,a)} = \alpha_{(p,t),(0,0)}^{(\ell,a)} \mathbf{v}_{(0,0)}^{(\ell,a)} + \sum_{p'=1}^{N} \sum_{t'=1}^{F} \alpha_{(p,t),(p',t')}^{(\ell,a)} \mathbf{v}_{(p',t')}^{(\ell,a)}$$

Divided Space-Time Attention

Concatenation of attention-weighted vectors from all the heads is concatenated, projected, added with residual connection from previous layer.

$$\mathbf{z'}_{(p,t)}^{(\ell)} = W_O \begin{bmatrix} \mathbf{s}_{(p,t)}^{(\ell,1)} \\ \vdots \\ \mathbf{s}_{(p,t)}^{(\ell,\mathcal{A})} \end{bmatrix} + \mathbf{z}_{(p,t)}^{(\ell-1)}$$

A projection is computed of this representation by passing the LayerNormed output through an MLP, again adding a residual connection.

$$\mathbf{z}_{(p,t)}^{(\ell)} = \mathrm{MLP}\left(\mathrm{LN}\left(\mathbf{z'}_{(p,t)}^{(\ell)}\right)\right) + \mathbf{z'}_{(p,t)}^{(\ell)}$$

Thus, we obtain the representation after the application of temporal attention.

Divided Space-Time Attention

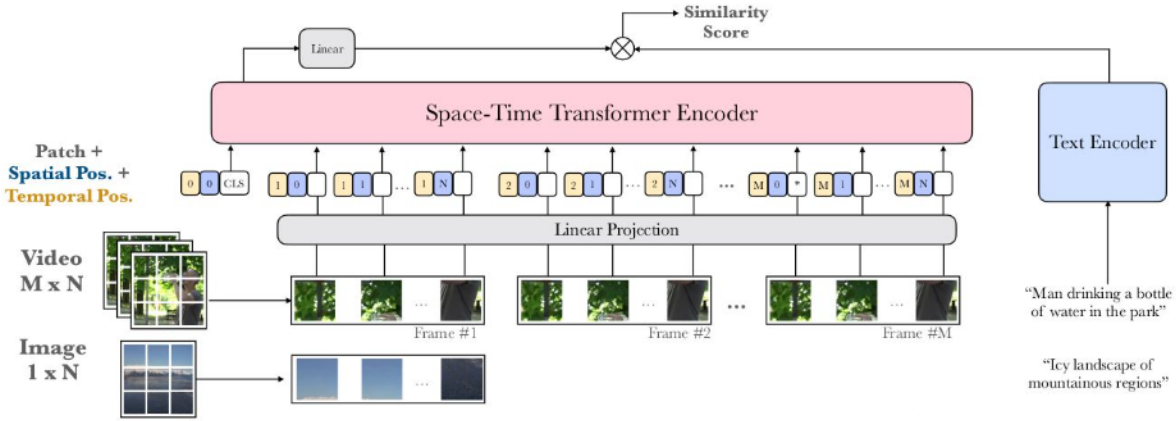
Spatial attention is computed over the representation given by temporal attention. Here, the query-key comparisons are performed for a patch (p, t) over all the patches at the same temporal instance.

$$\boldsymbol{\alpha}_{(p,t)}^{(\ell,a)\text{space}} = \text{SM}\left(\frac{\mathbf{q}_{(p,t)}^{(\ell,a)\top}}{\sqrt{D_h}} \cdot \left[\mathbf{k}_{(0,0)}^{(\ell,a)} \left\{\mathbf{k}_{(p',t)}^{(\ell,a)}\right\}_{p'=1,\ldots,N}\right]\right)$$

The subsequent steps are the same as for temporal attention till we obtain the z vector for spatial attention. The special embedding learnt at (0, 0) position is representative of the entire video representation across space & time.

Apart from reducing the computation of joint self-attention, the TimeSFormer architecture was found to have much less compute load at training and less inference cost as well.

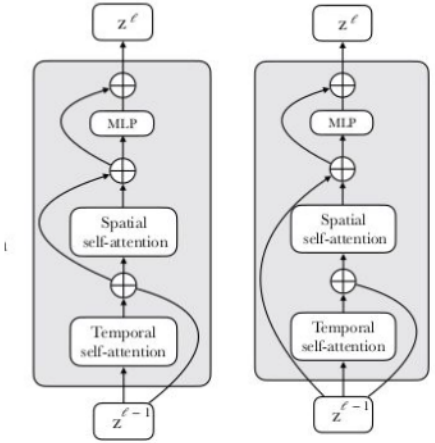
Frozen in Time: A Joint Video and Image Encoder for End-to-End Retrieval, ICCV 2021



This work introduces a dual encoder architecture suitable for retrieval. The video encoder is a TimeSFormer architecture, while the text encoder is a BERT-based model. The training involves joint image-video training, benefiting larger pretraining, where images are treated as videos 'frozen' in time. The model is trained using batch-wise contrastive learning for text-to-video and video-to-text mapping.

More on Frozen

- A minor modification in the computation of divided space-time attention involved replacement of the residual connection between temporal & spatial attention to that between input & spatial attention.
- The spatial attention weights were initialized from ViT, while the temporal ones were initialized with zero.
- They propose a curriculum learning strategy starting with images and gradually increasing the number of frames, with the intuition that the model gradually learns to attend to increasing temporal information.
- A new dataset for pretraining, WebVid-2M consisting of 2.5M video-text pairs was also introduced having an open domain & having larger, better text-aligned videos than Howto100M.



Divided space-time attention

Modification in Frozen

Training Frozen: Without using pretrained weights

Frozen was set to train for 100 epochs using a batch size of 8 and other parameters being kept the same as those used by the authors. The model achieved the best checkpoint at epoch 18. Results on test set are shown below.

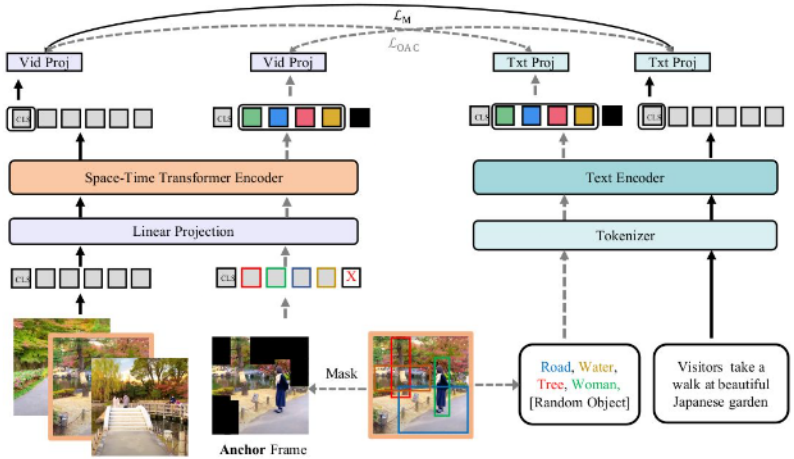
Recall			Rank	
1	5	10	Mean	Median
12.0	33.8	47.2	46.2	12

Training Frozen: Using pretrained weights

The pretrained weights of the model trained on the Conceptual Captions and WebVid-2M dataset were provided by the authors. They were used to to finetune on MSR-VTT for text-to-video retrieval. The model was trained for 5 epochs using a batch size of 8, learning rate of 3e-5 and AdamW optimization.

Recall			Rank	
1	5	10	Mean	Median
22.3	47.9	60.6	36.0	6.0

Object-aware Video-language Pre-training for Retrieval, CVPR 2022



Object-aware transformer (OATrans) is a dual-encoder pipeline, with ViT-based TimeSFormer for visual and DistilBERT for text modalities. The novelty of this architecture lies in using object regions and object tags in the input to visual and text streams. The model is trained using Object-Aware Contrastive (OAC) loss in addition to video-to-text and text-to-video matching loss objectives.

More on OATrans

- Anchor frame is introduced for combining the object information from all the sampled frames, in which the top K unique object categories are kept unmasked, while masking rest of the regions. Effectively serves as a single object image.
- Objects are extracted using Faster RCNN, capable of detecting 1600 object tags from Visual Genome.
- The OAC objective aligns these pairs of inputs:
  - raw videos to object tags
  - anchor frame to text
- The matched pairs in OAC are chosen so that the matching from object tags to anchor frame could end up learning a trivial function, disturbing the training.
- The model is considered to benefit from high-level semantics introduced by objects due to the inclusion of object information in visual & text encoders.

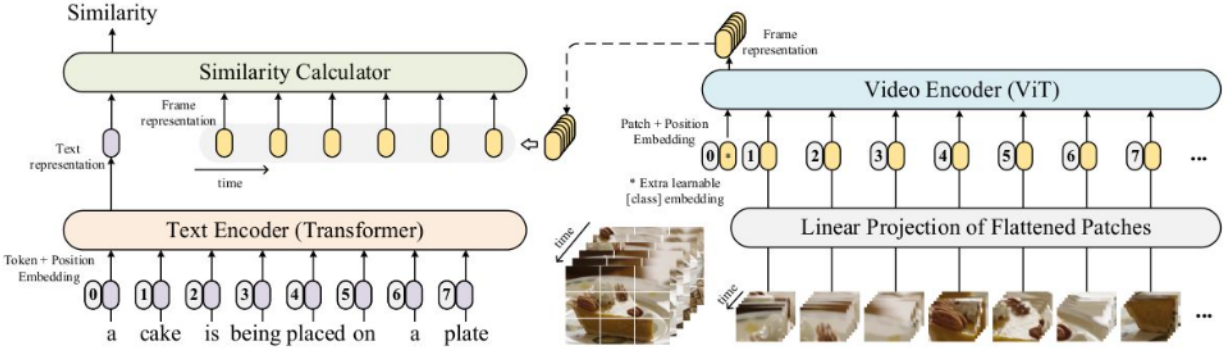
Training OATrans

OATrans was trained on the train split of MSRVTT dataset for 100 epochs with a batch size of 8 on 2 GPUs, while keeping the other hyperparameters the same as the authors. The training continued till the 100th epoch, the results on which are shown below.

Recall				Rank	
1	5	10	50	Mean	Median
12.8	35.6	48.1	75.5	52.6	12

Pretrained weights were not available, hence couldn't reproduce the results.

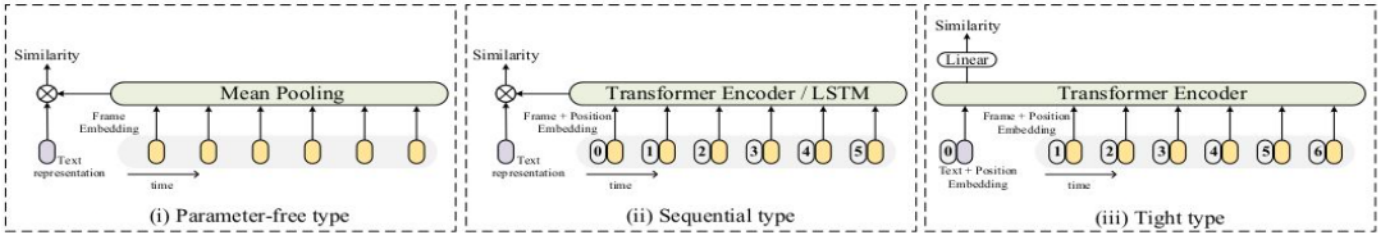
CLIP4Clip: An Empirical Study of CLIP for End to End Video Clip Retrieval, arXiv 2021



This work looks at leveraging the weights of powerful image-text CLIP by finetuning for video-text retrieval. Re-using the image and text encoder from CLIP, the model is post-pretrained on Howto100M dataset using MIL-NCE objective to have the model learn temporal features in videos. Finetuning using symmetrical contrastive loss on text-to-video and video-to-text objectives on 5 datasets yields SoTA performance.

More on CLIP4Clip

- For the input video, the default 2D projection from ViT performed better than 3D projection from ViViT (video vision transformer), perhaps because CLIP being trained heavily on images, couldn't adapt to 3D input.
- Three similarity functions were experimented with: parameter-less cosine similarity on mean pooling, cosine similarity on sequentially encoded input and similarity score on tightly encoded image and text input combined.
- Parameter-less mean pooling performed the best on lesser sized data, and was competitively good on larger sizes as well.
- Zero-shot performance of the post-pretrained model was noteworthy.



Finetuning on CLIP4Clip

The pretrained ViT-B/32 back bone of the CLIP model was used with a 2D linear patch for taking in video features. A mean pooling of video features is performed at the video encoder output to compute the cosine similarity with the text embedding. The model was trained for 5 epochs using a batch size of 16 and learning rate of 1e-4.

Recall			Rank	
1	5	10	Mean	Median
43.1	69.5	81.0	11.9	2.0

Summary results for trained video-text models

Model	Recall			Rank
	1	5	10	Median
HowTo100 (Zero shot)	7.50	21.20	29.60	38.0
HowTo100M (Finetuning)	17.40	41.00	55.60	8.0
UniVL (Finetuning)	22.50	51.40	66.30	5.0
Frozen (Without pretrained weights)	12.0	33.8	47.2	12.0
Frozen (With pretrained weights)	22.3	47.9	60.6	6.0
OATrans (Without pretrained weights)	12.8	35.6	48.1	12.0
CLIP4Clip (Finetuning)	**43.1**	**69.5**	**81.0**	**2.0**

Analysis on the performance obtained

- CLIP4Clip heavily outperforms all the other models. This shows the superiority of a model pretrained on enormous amount of data. Whether the pretrained architecture is image-based or video-based doesn't affect.
- The performance of OATransformer on non-pretrained weights has an edge over that of Frozen. This shows an improvement due to the addition of object detection features at input.
- The performance of finetuned UniVL has an edge over that of Frozen. This could indicate that a classical transformer-based modeling is sufficient than using space-time attention on videos. Or, this could correspond to a trade-off between joint video-text encoding (performance) v/s separate video and text encoders (efficiency).

Comparing index-based retrieval performance with the best & worst of embedding-based models

method	Recall									Rank	
	1	5	10	20	50	100	200	500	1000	Mean	Median
with paraphrase (nucleus)	2.17	8.16	12.98	19.73	28.73	37.09	48.80	61.57	73.71	660.68	212.5

Model	Recall			Rank
	1	5	10	Median
HowTo100 (Zero shot)	7.50	21.20	29.60	38.0
CLIP4Clip (Finetuning)	43.1	69.5	81.0	2.0

On using index-based retrieval as a primary stage in video retrieval

Considering the impreciseness in simpler index-based retrieval but higher efficiency in search, it can be used as an initial stage before application of embedding-based retrieval. However, the challenges based on results:

1. The zero-shot performance on the baseline model is at least 3 times better than the index-based model.
2. The amount of filtered videos to be taken from the output of index-based search has to be really high to ensure optimal use of the embedding-based model: R@1000 for the current index-based model is easily surpassed by R@10 of the SoTA embedding-based model.

Conclusion & Future Scope

- There is substantial gap between indexing-based retrieval of videos v/s SoTA which is based on embeddings, due to large semantic information being captured in the embedding space
- Improvement in video captioning can improve index-based video retrieval. The improvement can involve training the captioning model to generate multiple paraphrased versions guided by the video that could cover multiple possible query inputs to the model.
- Two prominent ways in which performance can be pushed:
  - Leveraging large-scale image-text pretraining from CLIP
  - Including object features in encoding

Thank you!