MDMMT: Multidomain Multimodal Transformer for Video Retrieval

Maksim Dzabraev[1,2], Maksim Kalashnikov[1], Stepan Komkov[1,2], Aleksandr Petiushko[1,2]
[1]Lomonosov Moscow State University, [2]Huawei Moscow Research Center
dzabraev.maksim@intsys.msu.ru, kalashnikov.maxim@intsys.msu.ru,
stepan.komkov@intsys.msu.ru, petyushko.alexander1@huawei.com

Abstract

We present a new state-of-the-art on the text-to-video retrieval task on MSRVTT and LSMDC benchmarks where our model outperforms all previous solutions by a large margin. Moreover, state-of-the-art results are achieved using a single model and without finetuning. This multidomain generalisation is achieved by a proper combination of different video caption datasets. We show that our practical approach for training on different datasets can improve test results of each other. Additionally, we check intersection between many popular datasets and show that MSRVTT as well as ActivityNet contains a significant overlap between the test and the training parts. More details are available at https://github.com/papermsucode/mdmmt.

1. Introduction

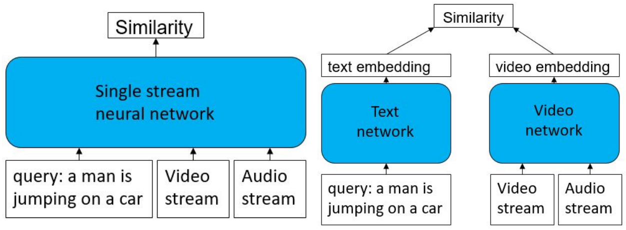
Video is a quite popular data format. More than 500 hours of video are uploaded on YouTube every minute. Personal mobile phones store gigabytes of video. Since video format gets more popular every year, the importance of modern search methods increases as well.

In this work we present our research on text-to-video retrieval task. In this task, system should return the most relevant video segments for an input textual query. The query is a textual description of what we want to find in the video gallery. The query may describe objects, actions, sounds and relations between them.

There are two major approaches to evaluation of the relevance between a textual search query and a video segment. The first approach is single-stream methods [28]. Here, a single network processes queries and videos simultaneously. The schematic illustration of this approach is depicted in Fig. 1a.

This type of approaches has an access to all input data from the beginning, thus they can produce accurate estimation of the relevance. Unfortunately, these approaches have a significant drawback, they are not scalable. For each input query, the search system should calculate the full forward pass for every video from the gallery.

The second approach is two-stream neural networks [20, 7]. Here a textual query and a video are processed by two different neural networks. As a result, the networks produce embeddings inside the same embedding space, where semantically similar textual queries and video segments are close to each other. The schematic illustration is depicted in Fig. 1b.



(a) Scheme for a single-stream neural network.

(b) Scheme for a two-stream neural network.

Figure 1: Two types of fusion

Two-stream models are scalable: they allow to precompute video embeddings for all videos from the gallery. Thus we can rapidly obtain relevances to all videos from the gallery. We need to run one forward pass of the textual network and compute the cosine similarity between the new query embedding and all precomputed video embeddings.

To make a strong video retrieval solution, it is important to show to the model a lot of scenes, actions and objects from the real life. Although there are a lot of datasets, however none of them covers all the aspects of life. To address this problem, we need to formulate rules for combining different existing datasets into one large training dataset.

To train a text-to-video retrieval neural network, the training dataset should consists of pairs: (a video segment, a textual description of this video segment). There is a number of video captioning datasets with similar structure of data that we can use for the text-to-video retrieval task also [35, 14, 25, 1, 15, 2, 38, 9, 27, 19].

The most common datasets for text-to-video retrieval are MSRVTT [35], ActivityNet [14] and LSMDC [25]. Nowadays, these datasets are quite popular for evaluation of solutions for text-to-video retrieval.

One of the first works addressed to text-to-video retrieval is [30]. One of the most universal solution for video retrieval task is Multi Modal Transformer [7]. It uses BERT [3] for encoding textual queries and the transformer encoder backbone [33] for encoding videos. The transformer encoder allows to process temporal dependencies inside the multi modal data source in a natural way.

Our work is based on the MMT approach and their codebase. We use all datasets described above for training.

Our main contributions in this work are the following:

- We present a new state-of-the-art (SotA) result on MSRVTT and LSMDC benchmarks;
- We present a model that shows good results on three different benchmarks at the same time without finetuning: MSRVTT (SotA), LSMDC (SotA) and ActivityNet;
- We present a practical approach which helps us to find the overlap between the training and the test parts of datasets.

2. Related Work

2.1. Datasets

MSRVTT [35] is traditionally used by researchers as the main dataset for testing text-to-video retrieval models. This dataset consists of 10k video segments. Each segment is described by 20 captions. The authors collect 257 popular search queries and gather 118 most relevant videos from YouTube for each of them. The dataset consists of 42 hours of videos. The captions are made by 1327 amazon workers.

There are three different training/test splits that are commonly used. The official split is called **full** split, where the training part consists of 7k videos and the test part consists of 3k videos. There are two important properties of this split: there are no two video segments from the same video that belong either to training part, either to test part; there are no two video segments retrieved from the same query that belong either to training part, either to test part.

Other two splits are called **1k-A** [36] (sometimes called jsfusion) and **1k-B** [18] (sometimes called miech). Both of them consist of different 1k videos for testing. They are created by random sampling of 1k videos from the original test part (full split). 1k-A training part consists of the full split training part and the rest of the videos from the test part, so it has 1k videos for the test part and 9k videos for the training part. 1k-B consists of 1k videos for the test part and 6.5k videos for the training. Both splits use only one caption per segment (instead of 20 captions for the full split).

Unfortunately 1k-A and 1k-B mix up the the original training and test parts. This led to violation of properties described for the full split.

Another problem is that all these splits have the overlap (overlap by content, not by YouTube ID) between the test and training parts, see C.2 for details. To be strict we remove the overlap between the test part and the training part of MSRVTT full split. We called this split MSRVTT **full clean**, and refer to it as M$_c$. It is worth to mention that we do not modify the test part, we remove some videos from the training part only.

The Large Scale Movie Description Challenge (LSMDC) [25] is the extension of two independent datasets: MPII Movie Description Dataset (MPII-MD) [24], and Montreal Video Annotation Dataset (M-VAD) [29].

Video segments for this dataset are cropped from movies, where movie textualized transcriptions are used as captions. A movie transcription is an audio description of a video segment that helps blind people to watch movies by description of what is happening, who appears in this time, what is on background right now and so on.

In this work for testing we use LSMDC public test, which consists of 1k video segments.

ActivityNet captions dataset [14] consists of 20k videos and 100k captions, where captions cover the full video length for the most of videos, and neighbour captions may intersect. The annotations are made with Amazon Mechanical Turk.

The situation when some video segments may overlap makes a problem for text-to-video retrieval testing. Suppose we have two video caption pairs (S_1, C_1) and (S_2, C_2) where the video segment S_1 has a non empty overlap with the video segment S_2. Now suppose that for query C_1 the system returns the video segment S_2. Is it mistake or not? What to do in this case?

Many previous works use ActivityNet test dataset in a paragraph retrieval mode. In this mode, all captions for all video segments are concatenated, then the concatenated text is used as a textual query and the whole video should be retrieved for this query. This mode has two drawbacks. The first one is that paragraph retrieval is not a classical video retrieval mode. It is an another task. One can ask: if a model is good in paragraph retrieval will it be good for video retrieval? The second drawback is that queries are long and video segments are long (compared to a classical video retrieval mode). This issue requires to enlarge the input for the model.

Another way to use the test part of ActivityNet is to sample a single random segment once per video. As a result we obtain non-intersected video segments and captions with usual length. We use ActivityNet test part in this way. We take all videos from val1 and val2 parts, and sample a single random segment from each video. All results on ActivityNet are reported on this split.

Additionally, in this work the following datasets are used: NIST TRECVID Twitter vines [1], TGIF [15], MSVD [2], YouCook2 [38], Something-something V2 [9], Kinetics 700 [27], HowTo100M [19].

2.2. Prior Art

A dominant approach to train video retrieval models is contrastive learning. The idea of this approach is that we have a set of pairs $(\text{video}_i, \text{text}_i)$ and elements of each pair should be placed next to each other in some metric space: $\text{distance}(\text{video}_i, \text{text}_i) \approx 0$ and at the same time the element video_i should be far from all other $\text{text}_j), j \neq i$: $\text{distance}(\text{video}_i, \text{text}_j) \gg 0$. The bi-directional max-margin ranking loss represents this idea [11].

When training data have a lot of noise, the MIL NCE loss can be applied in the training procedure [17]. Suppose that we know that a video_i should be close to one of (or several) texts $\{\text{text}_{i1}, ..., \text{text}_{ik}\}$. This approach tries to reduce the distance between the video_i and all $\{\text{text}_{i1}, ..., \text{text}_{ik}\}$ at the same time.

All video captions datasets have the following problem. Suppose the distance between $(\text{video}_i, \text{text}_i)$ is to be minimized while the distance between $(\text{video}_i, \text{text}_j), j \neq i$ is to be maximized, but text_i and text_j are quite similar from the semantical point of view. Maybe the optimal scenario in this situation is to minimize the distance between $(\text{video}_i, \text{text}_j), j \neq i$. In [21] the authors show the approach which deals with this problem.

As far as an input video is the temporal sequence of tokens (frames or video segments) it is important to efficiently aggregate the information from all tokens. Many ideas for such aggregation in the previous works are borrowed from the natural language processing. Convolution filters for aggregation are used in [21], a transformer encoder as a video aggregator is used in [7], many different aggregation functions are tested in [22].

We think that the most promising aggregation method is a Multi Modal Transformer (MMT) method [7]. MMT is a two-stream solution designed for a text-to-video retrieval task. The extraction of features from the input video stream is done in the following way. An input video is preprocessed by several pretrained frozen neural networks (these networks are called experts). Original solution uses seven modalities: motion, RGB, scene, face, OCR, speech, audio. One pretrained network for each modality is used. The motion modality is processed with video recognition networks like S3D [34], SlowFast [6], irCSN [8], where several input frames are used as a single input. The RGB modality uses a single frame as an input. The audio modality uses the raw input sound from a video. After embeddings are extracted from input data by these experts, it will be augmented by adding positional encoding tokens (representing time) and expert tokens. Then the augmented embeddings are passed through the MMT backbone. the MMT backbone is a standard transformer encoder architecture. Each input modality produces one embedding, so in total there are seven output embedding from MMT.

For encoding the textual query the authors use pretrained BERT model where the output [CLS] token is used. The output is postprocessed with shallow networks (one network per modality) to extract the modality related information. In total seven feature vectors are produced. In addition to embeddings produced from the text query, seven weights are computed. This weights characterize how much the query describes one of seven modalities. For example, if a query does not represent the sound, the small weight for the audio modality should be produced.

The final similarity score is done by a sum of seven weighted dot products of embeddings.

The MMT is trained with the bi-directional max-margin ranking loss [11]:

$$\frac{1}{B}\sum_{i=1}^{B}\sum_{j\neq i}\Big[\max(0, s_{ij}-s_{ii}+m)+\max(0, s_{ji}-s_{ii}+m)\Big] \tag{1}$$

where B represents the batch size, s_{ij} is the similarity between the i-th query and the j-th video inside this batch, and m is some predefined margin correspondingly.

3. Methodology

Our work is mostly based on MMT. We use the same loss and a similar architecture, but with different hyperparameters. In this work we study the following questions: Which publicly available pretrained motion expert is the best for text-to-video retrieval, Sec. 3.1. How to combine several video caption datasets in order to train a strong model without specialisation for a particular dataset, Sec. 3.2. The problem about intersection of test and training parts of several datasets is discussed in Supplementary Sec. C.

3.1. Motion Experts

The MMT video backbone does not process the raw input video stream. Instead, the input video stream is processed by one or more pretrained experts, where each expert produces time series of features. The most important modality is motion: a motion expert processes several video frames as a single input unit and extracts the information about actions and objects within a segment.

We may say that the motion modality is the basis of the MMT. If a motion expert does not extract some information then there is a high probability that MMT does not know about some events in the video stream. That is why improvement the motion expert is very important.

We consider several best solutions from Kinetics [13] benchmark as well as several promising video recognition models and check which one works in the best way as a motion expert. We present all details in Sec. 4.2.

3.2. Dataset Creation

It is possible to train a video retrieval model by two means. The first way is the way of specialization for a single domain. For example: create model that will work well only for MSRVTT benchmark (or domain) but at the same time this model will show poor results on other datasets (domains). In this way MMT [7] is trained. The authors trained three different models for MSRVTT, ActivityNet and LSMDC datasets. Each of these three networks works good on domain X if and only if it is trained on X, but at the same time works poorly on another domain $Y \neq X$. We provide a proof of this statement in Tab. 6.

The second way is to create a model that works good for all domains at the same time. We use this way.

Obviously, the model trained in the first way can not work well with real users, because we can not guarantee that users will use captions similar to captions of a small training dataset.

The second drawback here is that each video retrieval training dataset is not that big, and it causes the problem the model does not see many words and real life situations during training. For example, MSRVTT has only 9k videos and 200k captions in total for training. Obviously, this is not enough to train a neural network that will know most of real life situations, different items and persons. To tackle with this problem we can take several datasets with videos and captions and concatenate them.

Different datasets have different numbers of videos and the different number of captions per video; Some datasets may have long captions and some may have short captions; Different rules for creating captions are used by human writers, and so on. Due to these factors, some datasets may contain more information and require longer training time and some datasets may contain less information and require shorter training time. On the other hand, if we use long training time for a small dataset, it could lead to overfitting on this dataset (the data may be memorized). The "information sizes" of some used datasets are depicted in Fig. 2.

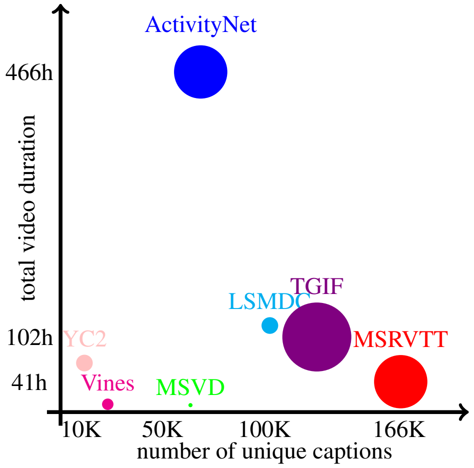


Figure 2: Radius of the ball represent the "information size" of dataset. The biggest balls have more diversity in data.

Fig. 2 is made with a simple algorithm. First, we take the original training procedure of MMT. Then, for a given dataset, we change the number of examples that will be shown to a network during training. We define the radius of the ball as the number of training examples for which the performance gets saturated (*i.e.* increasing of the training time does not give the better model).

The key question is: what is the proper way for sampling examples from several datasets taking into account the different information size?

We use these simple rules: 1. If a dataset X is larger than Y, we should sample from X more often than from Y; 2. Training on X and Y combined requires longer train than training solely on X or Y; 3. Training on X and Y combined may require a deeper model than for X or Y.

Our experiments show that the proper usage of rules 1–3 often improves the results for a specific test dataset (*e.g.* MSRVTT) after extending the training dataset.

We managed to combine the following datasets: MSRVTT, ActivityNet, LSMDC, TwitterVines, YouCook2, MSVD, TGIF and Something-to-Something V2 (SomethingV2). In total, we increase the number of video segments by 40 times and the number of unique captions by 4 times compared with MSRVTT dataset. In Tab. 1 we summarize the sizes of used datasets. We separate SomethingV2 dataset from all other datasets because: 1. all video segments are created artificially, 2. the structure of text captions is quite limited. At the same time videos for all other datasets are collected from the Internet and captions created by humans have quite a rich structure.

Dataset	Num video	Num pairs	Num unique captions	Has YouTube Id
MSRVTT	10k	200k	167k	Yes
ActivityNet	14k	70k	69k	Yes
LSMDC	101k	101k	101k	No
TwitterVines	6.5k	23k	23k	No
YouCook2	1.5k	12k	12k	Yes
MSVD	1.5k	80k	64k	Yes
TGIF	102k	125k	125k	No
Sum above	**236k**	**611k**	**561k**	—
SomethingV2	193k	193k	124k	No
Sum above	**429k**	**804k**	**685k**	—

Table 1: The "Num video" column represents the number of video clips in the dataset, the "Num pairs" column represents the total number of video caption pairs, the "Num unique captions" column represents the number of unique captions in the dataset.

3.3. Intersection

It is important to extend the training dataset carefully. We should not allow the intersection among video segments in training and test parts.

To find the intersection between the test part and the training part, we use the two stage filtration. The first stage is to use the YouTube ID. If it is available. We should not allow to use two video segments sampled from the same video in the test and training parts simultaneously.

In the second stage, we compute the similarity score between each video from the test part and each video from the training part and then we manually assess the pairs with the highest scores. In total we assess more than 100K pairs of the most relevant segments.

We find that about 37% of video segments from the MSRVTT 1k-A test part have a pair with the same YouTube ID in the MSRVTT 1k-A training part (these segments may not overlap). For the MSRVTT 1k-B split, about 38% of video segments from the test part have a pair in the training part with the same YouTube ID. We do not find intersection by embeddings between test and training parts of 1k-A and test and training parts of 1k-B splits. the MSRVTT full split does not have intersection between training and test parts by YouTube ID. Using filtration by embeddings we find that about 3% of video segments in the test part have a pair in the training part. For ActivityNet (using intersection by embeddings) we find that about 3% of video segments from the validation part have pair in the training part.

Our approach allows to approximately estimate the total number of videos in the intersection without finding the exact intersection. Using it, we estimate (but do not find) the overlap between HowTo100M and MSRVTT, and conclude that about 10% of the MSRVTT full test may be in the HowTo100M dataset. We make similar estimation for ActivityNet and Kinetics700. Estimation shows that about 10% or more of the ActivityNet validation may be in the Kinetics 700 dataset.

The details about filtration algorithm and its results are presented in Supplementary Sec. C.

4. Experiments

4.1. Architecture

We use exactly the same neural network architecture as original MMT [7]. Our method is based on their codebase. The difference is in the following: 1. we use the more aggressive dropout that equals to 0.2 for the text based BERT and the video based transformer encoder (against the original value of 0.1); 2. we observe that the deeper and wider transformer encoder for a video network gives better results — we use 6 layers and 8 heads for the motion only modality and 9 layers and 8 heads for the motion + audio setting (against 4 layer and 4 head in the original implementation).

4.2. Stronger Motion Experts

As the input data for MMT is embeddings from experts, the question arises: if a better expert is used, will we have a stronger model? To answer this question, we train MMT on MSRVTT dataset with the only motion modality. For motion experts we try several architectures pretrained on different datasets. These models are presented in Tab. 3. We take the architectures that show the best results on the Kinetics 400 benchmark and that have publicly available pretrained weights: [34, 6, 31, 32, 8].

Abbreviate	Composition
M,$M_{1k\text{-}A}$,$M_{1k\text{-}B}$	MSRVTT full, 1k-A, 1k-B splits
M_c	MSRVTT full clean split
A	ActivityNet
A_{val1}, A_{val2}	ActivityNet val1, val2 validation sets
$A_{p/r}$	ActivityNet paragraph retrieval
L	LSMDC
K	Kinetics700
V	Twitter Vines
Y	YouCook2
HT100M	HowTo100M
MALV	MSRVTT + ActivityNet + LSMDC + TwitterVines
MALVYMT	MSRVTT + ActivityNet + LSMDC + TwitterVines + YouCook2 + MSVD + TGIF
MALVYMTS	MSRVTT + ActivityNet + LSMDC + TwitterVines + YouCook2 + MSVD + TGIF + Something to Something V2

Table 2: The left column represents the abbreviate name for the set of datasets from the right column. See details for MSRVTT full clean split and ActivityNet paragraph retrieval in Sec. 2.1.

The results in Tab. 3 are made with the same hyperparameters as in [7]. For the training dataset we use only MSRVTT full clean split. The first line in Tab. 3 represents the motion feature extractor from the original MMT paper.

As we can see, usually stronger models provide better results, but not always. Refer to r(2+1)d 152 rows, this network demonstrates one of the best performance on Kinetics 400 benchmark, but works poorly as motion expert. Maybe this network is over-specialized for Kinetics 400. More shallow analogue of r(2+1)d 152 is r(2+1)d 34 which shows much better results.

An interesting observation is that the best results are achieved with the networks trained in the unsupervised manner. CLIP and models trained on IG65M outperform all other models trained on Kinetics in the supervised manner. Another weakly supervised dataset is Sports1M [12]. Models trained on this dataset provide weak embeddings similar to the weak s3d model trained on Kinetics dataset. The CLIP [23] (ViT-B/32) image feature extractor outperforms all other models with a large margin. The model s3dg MIL-

Video expert	Dataset	MSRVTT full clean Text → Video				
		R@1↑	R@5↑	R@10↑	MnR↓	MdR↓
s3d	Kinetics 600	$7.7_{\pm0.1}$	$24.0_{\pm0.2}$	$34.9_{\pm0.2}$	$129.6_{\pm1.0}$	$23.7_{\pm0.5}$
SlowFast 32x2 R101	Kinetics 600	$9.3_{\pm0.1}$	$27.5_{\pm0.1}$	$39.1_{\pm0.1}$	$110.8_{\pm1.1}$	$18.7_{\pm0.5}$
ipCSN152	IG65M	$9.5_{\pm0.1}$	$27.9_{\pm0.2}$	$39.6_{\pm0.2}$	$106.1_{\pm1.1}$	$18.0_{\pm0.0}$
ipCSN152	IG65M → K400	$8.3_{\pm0.1}$	$25.2_{\pm0.1}$	$36.5_{\pm0.2}$	$124.3_{\pm0.2}$	$21.0_{\pm0.0}$
ipCSN152	Sports1M	$7.4_{\pm0.2}$	$22.4_{\pm0.1}$	$32.7_{\pm0.2}$	$140.6_{\pm1.0}$	$27.0_{\pm0.0}$
ipCSN152	Sports1M → K400	$7.8_{\pm0.1}$	$24.2_{\pm0.1}$	$35.2_{\pm0.1}$	$129.9_{\pm0.2}$	$23.0_{\pm0.0}$
irCSN152	IG65M	$9.5_{\pm0.1}$	$27.9_{\pm0.2}$	$39.5_{\pm0.2}$	$105.5_{\pm0.4}$	$18.0_{\pm0.0}$
irCSN152	IG65M → K400	$8.4_{\pm0.1}$	$25.3_{\pm0.1}$	$36.5_{\pm0.2}$	$120.4_{\pm0.4}$	$21.0_{\pm0.0}$
irCSN152	Sports1M	$6.9_{\pm0.1}$	$21.6_{\pm0.1}$	$31.6_{\pm0.1}$	$141.9_{\pm0.4}$	$28.7_{\pm0.5}$
irCSN152	Sports1M → K400	$7.7_{\pm0.1}$	$24.1_{\pm0.1}$	$35.1_{\pm0.1}$	$127.6_{\pm0.6}$	$23.0_{\pm0.0}$
r(2+1)d 152	IG65M	$5.7_{\pm0.1}$	$18.5_{\pm0.1}$	$27.8_{\pm0.1}$	$178.5_{\pm1.5}$	$37.7_{\pm0.9}$
r(2+1)d 152	IG65M → K400	$5.5_{\pm0.1}$	$18.1_{\pm0.1}$	$27.3_{\pm0.1}$	$184.1_{\pm1.2}$	$39.3_{\pm0.5}$
r(2+1)d 152	Sports1M → K400	$5.3_{\pm0.1}$	$17.3_{\pm0.1}$	$26.0_{\pm0.1}$	$193.4_{\pm3.6}$	$42.3_{\pm0.5}$
r(2+1)d 34	IG65M	$9.1_{\pm0.2}$	$27.2_{\pm0.2}$	$38.7_{\pm0.2}$	$108.1_{\pm0.0}$	$19.0_{\pm0.0}$
r(2+1)d 34	IG65M → K400	$8.2_{\pm0.2}$	$25.3_{\pm0.3}$	$36.7_{\pm0.1}$	$120.8_{\pm0.7}$	$21.0_{\pm0.0}$
CLIP	CLIP	$\mathbf{14.4}_{\pm0.1}$	$\mathbf{37.4}_{\pm0.3}$	$\mathbf{50.2}_{\pm0.3}$	$\mathbf{70.3}_{\pm0.3}$	$\mathbf{10.3}_{\pm0.5}$
s3dg MIL-NCE	HowTo100M	$8.6_{\pm0.4}$	$26.3_{\pm0.5}$	$37.9_{\pm0.7}$	$104.4_{\pm2.2}$	$19.3_{\pm0.5}$

Table 3: Comparison of the best available pretrain models as the motion experts for MMT. IG65M → K400 means that model is trained on IG65M and then fine tuned on Kinetics400.

NCE is a video encoder from the work [17]. This network is trained from scratch on HowTo100M dataset.

As we show in Supplementary Sec. C Kinetics dataset has an overlap with MSRVTT dataset, and we do not know whether it affects to overfitting or not. Also it is worth to mention that IG65M and CLIP datasets are not publicly available, so we do not know if there is an overlap with MSRVTT and other video retrieval datasets.

For more details about our usage of pretrained video experts refer to Supplementary Sec. A.

4.3. Datasets Combination

In this section we show our experiments about the combination of different datasets. Nowadays, video caption datasets are not big enough to capture all real life situations. Some datasets may be biased also. A combination of different datasets may help to tackle this problem.

Our experiments show that the proper combination of datasets allows to train a single model that can capture the knowledge from all used datasets. An important thing here is that in most cases the model trained on the combination of datasets is better than the model trained on a single dataset.

In our experiments we combine all datasets presented in Tab. 5. A important thing is how to sample minibatches during training. In our experiments we first sample a dataset, then we uniformly sample a video segment. If this sampled video segment has more than one caption then we sample a single caption uniformly. Column weight in Tab. 5 describes the probability of sampling the corresponding dataset. To obtain the probability of sampling the dataset with the weight w, we should divide w by the sum of all weights.

The weights for all datasets are manually adjusted. It is important to find a good weight combination. If some weight is larger than needed then this dataset is overseen and the performance result is lower in comparison to the optimal case. The opposite case is when a small weight is selected. This causes the situation when a network does not see the required number of examples from this dataset during training.

We use MMT with the motion modality only for experiments in this section. Embeddings for the motion modality are computed with irCSN152 pretrained on IG65M. All configurations are trained with 50 epochs and different number examples per epoch. The initial learning rate is 5e-5. After each epoch we multiply learning rate by 0.95. The MALVYMTS (see Tab. 2 for abbreviations) configuration is trained with 150K examples per epoch. Configurations with the less number of datasets are trained with the less number of examples per epoch. The number of examples per epoch can be represented as a product of 150K by a sum of normalized weights (weights from Tab. 5 divided by a sum of all weights) for each dataset (the initial sum equals to 1): $150K = 150K \times (p_{\text{MSRVTT}} + p_{\text{ActivityNet}} + p_{\text{LSMDC}} + p_{\text{Twitter Vines}} + p_{\text{YouCook2}} + p_{\text{MSVD}} + p_{\text{TGIF}} + p_{\text{Something V2}})$. If some dataset is removed from the training then we remove the corresponding coefficient from this sum, so the resulting length will be 150K multiplied by a value less than 1.

As far as we use the configurations M$_c$, A, L as the baselines, we need to be sure that the results for these config-

	model	ActivityNet text → video				
		R@1↑	R@5↑	R@10↑	MnR↓	MdR↓
	MMT ($A_{p/r}$) motion+audio [7]	7.3	22.5	31	283.9	30
	CLIP [23]	9.4	22.8	31.3	302.3	35
Ours	MDMMT(M_cALVYMTS)	**20.1**$_{\pm 0.5}$	**45.1**$_{\pm 0.5}$	**58.0**$_{\pm 0.6}$	**70.8**$_{\pm 0.1}$	**7.0**$_{\pm 0.0}$

Table 4: Test results on our split (see Sec. 2.1) on ActivityNet.

urations are the optimal values. In addition to the rule described above, we try several values for a number of examples per epoch parameter and report the results for the best found value.

Dataset	Weight
MSRVTT	140
ActivityNet	100
LSMDC	70
Twitter Vines	60
YouCook2	9
MSVD	9
TGIF	102
Something V2	169

Table 5: The ”Weight” column describes how often examples are sampled from the dataset.

Dataset	Test Text → Video R@5 ↑		
	MSRVTT	ActivityNet	LSMDC
M_c	29.0$_{\pm 0.2}$	*13.4*$_{\pm 0.3}$	*12.9*$_{\pm 0.6}$
A	*14.7*$_{\pm 0.1}$	30.9$_{\pm 0.6}$	*10.4*$_{\pm 0.3}$
L	*8.8*$_{\pm 0.1}$	*7.2*$_{\pm 0.2}$	24.7$_{\pm 0.6}$
M_cALV	32.1$_{\pm 0.1}$	32.0$_{\pm 0.2}$	26.5$_{\pm 0.7}$
M_cALVYMT	33.8$_{\pm 0.1}$	32.3$_{\pm 0.2}$	27.3$_{\pm 0.4}$
M_cALVYMTS	**34.5**$_{\pm 0.1}$	**32.4**$_{\pm 0.5}$	**27.4**$_{\pm 0.6}$

Table 6: The first three rows M_c,A,L report the quality of models trained on a single domain, and tested on other domains. *Italic* means that the model did not see data from this domain during training. Only motion modality (irCSN152) is used.

Tab. 6 summarizes our experiments on the datasets combination (graphical representation of Tab. 6 is given in Supplementary Sec. B). The main point here is that the proper combination of datasets leads to the best solution.

4.4. Final Results

In this section we compare our solution with the prior art. Our two best solution uses three modalities: the audio, the motion and RGB. To fuse modalities we use MMT architecture with 9 layers and 8 heads. As a feature extractor for the audio stream the vggish network [10] is used. For the video encoding we use CLIP ViT-B/32 (RGB modality) and irCSN152 (motion modality) pretrained on IG65M dataset. The details about preprocessing videos for both networks are presented in Supplementary Sec. A.

Additionally in Supplementary Sec. F we report separate results for motion + audio encoders and RGB + audio encoders because we do not know whether the IG65M or CLIP training dataset has a significant overlap with any of the test datasets or not.

All our models presented in Tab. 4, 7 and 8 are trained based on the pretrain HowTo100M model. We present the details about pretraining in Supplementary Sec. E.

The results for MSRVTT are presented in Tab. 7. As we can see, our solution MDMMT(MALVYMTS) L9H8 CLIP+irCSN152+audio significantly outperforms all previous solutions on all splits. Our solution is better than the previous SotA (on R@5) on 8.7% and 10.5% on full and 1k-A correspondingly. It is also worth to mention that our MDMMT (using only the motion, the RGB and the audio modalities) outperforms the original MMT (using the motion, the RGB and the audio and 4 other modalities) by 18.7% and 11.9% (R@5) on full and 1k-A correspondingly.

We also report the results for the original CLIP [23]. The CLIP model has an image encoder and a text encoder, both pretrained in an unsupervised way. To test the CLIP model we take a single frame from the middle of the video (this is the original testing protocol for CLIP). The row CLIP agg [22] represents the usage of CLIP model with several frames using some specific aggregation procedure from this work.

In Tab. 8 we report the results on LSMDC. On this benchmark we outperform the previous SotA solution by 8.6%.

As we mention in Sec. 2.1, we do not use the standard ActivityNet paragraph retrieval test protocol. Instead, we use the text-to-video retrieval protocol. To compare our solution with the previous work we tested two previous models: MMT trained on ActivityNet in paragraph retrieval mode and CLIP. The results are reported in Tab. 4. Our solution outperforms MMT by 22.6% and CLIP by 22.3%. The row MMT ($A_{p/r}$) motion+audio means that this network is trained only on ActivityNet dataset with the paragraph retrieval mode using motion and audio modalities.

The important property of our model is that we train a single model and test it on different test sets. The authors of previous SotA approach (MMT) trained three different models for MSRVTT, ActivityNet and LSMDC, while in

	model	split	MSRVTT text → video				
			R@1↑	R@5↑	R@10↑	MnR↓	MdR↓
	Random baseline	full	0.0	0.2	0.3	1500	1500
	VSE [20]		5.0	16.4	24.6	—	47
	VSE++ [20]		5.7	17.1	24.8	—	65
	Multi Cues [20]		7.0	20.9	29.7	—	38
	W2VV [4]		6.1	18.7	27.5	—	45
	Dual Enc. [5]		7.7	22.0	31.8	—	32
	CE [16]		$10.0_{\pm0.1}$	$29.0_{\pm0.3}$	$41.2_{\pm0.2}$	$86.8_{\pm0.3}$	$16.0_{\pm0.0}$
	MMT (M) 7mod [7]		$10.7_{\pm0.2}$	$31.1_{\pm0.1}$	$43.4_{\pm0.2}$	$88.2_{\pm0.7}$	$15.0_{\pm0.0}$
	CLIP [23]		15.1	31.8	40.4	184.2	21
	CLIP agg [22]		21.5	41.1	50.4	—	**4**
Ours	MDMMT(MALVYMTS)		**23.1**$_{\pm0.1}$	**49.8**$_{\pm0.1}$	**61.8**$_{\pm0.1}$	**52.8**$_{\pm0.2}$	$6.0_{\pm0.0}$
	Random baseline	1k-A	0.1	0.5	1.0	500.0	500.0
	JSFusion [36]		10.2	31.2	43.2	—	13
	E2E [17]		9.9	24.0	32.4	—	29.5
	HT [19]		14.9	40.2	52.8	—	9
	CE [16]		$20.9_{\pm1.2}$	$48.8_{\pm0.6}$	$62.4_{\pm0.8}$	$28.2_{\pm0.8}$	$6.0_{\pm0.0}$
	CLIP [23]		22.5	44.3	53.7	61.7	8
	MMT ($\mathrm{M}_{\text{1k-A}}$) 7mod [7]		$26.6_{\pm1.0}$	$57.1_{\pm1.0}$	$69.6_{\pm0.2}$	$24.0_{\pm0.8}$	$4.0_{\pm0.0}$
	AVLnet[26]		27.1	55.6	66.6	—	4
	SSB [21]		30.1	58.5	69.3	—	3.0
	CLIP agg [22]		31.2	53.7	64.2	—	4
Ours	MDMMT($\mathrm{M}_{\text{1k-A}}$ALVYMTS)		**38.9**$_{\pm0.6}$	**69.0**$_{\pm0.1}$	**79.7**$_{\pm0.6}$	**16.5**$_{\pm0.4}$	**2.0**$_{\pm0.0}$

Table 7: Results on MSRVTT dataset.

	model	LSMDC text → video				
		R@1↑	R@5↑	R@10↑	MnR↓	MdR↓
	CT-SAN [37]	5.1	16.3	25.2	—	46
	JSFusion [36]	9.1	21.2	34.1	—	36
	MEE [18]	9.3	25.1	33.4	—	27
	MEE-COCO [18]	10.1	25.6	34.6	—	27
	CE [16]	$11.2_{\pm0.4}$	$26.9_{\pm1.1}$	$34.8_{\pm2.0}$	$96.8_{\pm5.0}$	$25.3_{\pm3.1}$
	CLIP agg [22]	11.3	22.7	29.2	—	56.5
	CLIP [23]	12.4	23.7	31.0	142.5	45
	MMT (L) 7mod [7]	$12.9_{\pm0.1}$	$29.9_{\pm0.7}$	$40.1_{\pm0.8}$	$75.0_{\pm1.2}$	$19.3_{\pm0.2}$
Ours	MDMMT(M_cALVYMTS)	**18.8**$_{\pm0.7}$	**38.5**$_{\pm0.4}$	**47.9**$_{\pm0.7}$	**58.0**$_{\pm1.1}$	**12.3**$_{\pm0.5}$

Table 8: Test results on LSMDC public test (1k video)

Tab. 6 we show that the model trained in such a manner has poor generalization and can show good performance on the test part of the dataset X if and only if it is trained on the training part of the dataset X.

5. Conclusions and Discussion

In this work we present a new text-to-video retrieval state-of-the-art model on MSRVTT and LSMDC benchmarks. We do not use ActivityNet dataset in the paragraph retrieval mode as many previous works do, so we can not compare to them. However, we show that in the video retrieval mode on ActivityNet we outperform the previous state-of-the-art model (MMT) by a large margin. Our model captures knowledge from many video caption datasets. Thus it is able to show the best results on several datasets at the same time without finetuning.

We also present a practical approach to find the overlap between two different video datasets. Using this approach we find the overlap between several datasets. Especially, we find a large overlap between the MSRVTT test and training parts, and between the ActivityNet test and training parts. Removal of this overlap from the MSRVTT training part significantly decreases the performance of the previous best models on the MSRVTT benchmark.

References

[1] George Awad, Asad A. Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse

Zhang, Eliot Godard, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Qunot. Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *Proceedings of TRECVID 2020*. NIST, USA, 2020. 1, 2

[2] David Chen and William Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 190–200, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. 1, 2

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 1

[4] Jianfeng Dong, Xirong Li, and Cees G. M. Snoek. Predicting visual features from text for image and video caption retrieval. *IEEE Transactions on Multimedia*, 20(12):33773388, Dec 2018. 8

[5] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, Yuan He, Gang Yang, and Xun Wang. Dual encoding for zero-example video retrieval, 2019. 8

[6] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition, 2019. 3, 5

[7] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval, 2020. 1, 3, 4, 5, 7, 8

[8] Deepti Ghadiyaram, Matt Feiszli, Du Tran, Xueting Yan, Heng Wang, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition, 2019. 3, 5

[9] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzyska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The ”something something” video database for learning and evaluating visual common sense, 2017. 1, 2

[10] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification, 2017. 7

[11] Andrej Karpathy, Armand Joulin, and Li Fei-Fei. Deep fragment embeddings for bidirectional image sentence mapping. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 18891897, Cambridge, MA, USA, 2014. MIT Press. 3

[12] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014. 5

[13] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset, 2017. 3

[14] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *International Conference on Computer Vision (ICCV)*, 2017. 1, 2

[15] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. TGIF: A New Dataset and Benchmark on Animated GIF Description. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 1, 2

[16] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts, 2020. 8

[17] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos, 2020. 3, 6, 8

[18] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data, 2020. 2, 8

[19] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019. 1, 2, 8

[20] Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K Roy-Chowdhury. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, pages 19–27, 2018. 1, 8

[21] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, Joo Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning, 2021. 3, 8

[22] Jess Andrs Portillo-Quintero, Jos Carlos Ortiz-Bayliss, and Hugo Terashima-Marn. A straightforward framework for video retrieval using clip, 2021. 3, 7, 8

[23] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. *Image*, 2:T2. 5, 7, 8

[24] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description, 2015. 2

[25] Anna Rohrbach, Atousa Torabi, Marcus Rohrbach, Niket Tandon, Christopher Pal, Hugo Larochelle, Aaron Courville, and Bernt Schiele. Movie description, 2016. 1, 2

[26] Andrew Rouditchenko, Angie Boggust, David Harwath, Dhiraj Joshi, Samuel Thomas, Kartik Audhkhasi, Rogerio Feris, Brian Kingsbury, Michael Picheny, Antonio Torralba, and James Glass. Avlnet: Learning audio-visual language representations from instructional videos, 2020. 8

[27] Lucas Smaira, Joo Carreira, Eric Noland, Ellen Clancy, Amy Wu, and Andrew Zisserman. A short note on the kinetics-700-2020 human action dataset, 2020. 1, 2

[28] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning, 2019. 1

[29] Atousa Torabi, Christopher Pal, Hugo Larochelle, and Aaron Courville. Using descriptive video services to create a large data source for video annotation research, 2015. 2

[30] Atousa Torabi, Niket Tandon, and Leonid Sigal. Learning language-visual embedding for movie understanding with natural-language, 2016. 1

[31] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks, 2019. 5

[32] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition, 2018. 5

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017. 2

[34] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification, 2018. 3, 5

[35] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), June 2016. 1, 2

[36] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval, 2018. 2, 8

[37] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering, 2017. 8

[38] Luowei Zhou, Nathan Louis, and Jason J. Corso. Weakly-supervised video object grounding from text by loss weighting and object interaction, 2018. 1, 2