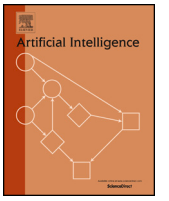
Artificial Intelligence 338 (2025) 104235

Contents lists available at ScienceDirect

Artificial Intelligence

journal homepage: www.elsevier.com/locate/artint



TEACHTEXT: CrossModal text-video retrieval through generalized distillation

Ioana Croitoru [a,b,1], Simion-Vlad Bogolin [a,b,1], Marius Leordeanu [c], Hailin Jin [d], Andrew Zisserman [a], Yang Liu [a,e,*], Samuel Albanie [f,**]

[a] *Visual Geometry Group, University of Oxford, United Kingdom*
[b] *Institute of Mathematics of the Romanian Academy, Romania*
[c] *University Politehnica of Bucharest, Romania*
[d] *Adobe Research, USA*
[e] *Wangxuan Institute of Computer Technology, Peking University, China*
[f] *Department of Engineering, University of Cambridge, United Kingdom*

ARTICLE INFO

Keywords:
Text-video retrieval
Distillation
Text embeddings
Video experts

ABSTRACT

In recent years, considerable progress on the task of text-video retrieval has been achieved by leveraging large-scale pretraining on visual and audio datasets to construct powerful video encoders. By contrast, despite the natural symmetry, the design of effective algorithms for exploiting large-scale language pretraining remains under-explored. In this work, we investigate the design of such algorithms and propose a novel generalized distillation method, TEACHTEXT, which leverages complementary cues from multiple text encoders to provide an enhanced supervisory signal to the retrieval model. TEACHTEXT yields significant gains on a number of video retrieval benchmarks without incurring additional computational overhead during inference and was used to produce the winning entry in the Condensed Movie Challenge at ICCV 2021. We show how TEACHTEXT can be extended to include multiple video modalities, reducing computational cost at inference without compromising performance. Finally, we demonstrate the application of our method to the task of removing noisy descriptions from the training partitions of retrieval datasets to improve performance. Code and data can be found at https://www.robots.ox.ac.uk/~vgg/research/teachtext/.

1. Introduction

The focus of this work is *text-video retrieval*—the task of identifying which video among a pool of candidates best matches a natural language query describing its content. Video search has a broad range of applications across domains such as wildlife surveillance, security, industrial process monitoring, and entertainment. Moreover, as humanity continues to produce video at ever-increasing scale, the ability to perform such searches effectively and efficiently takes on critical commercial significance to video hosting platforms such as YouTube.

* Corresponding author at: Visual Geometry Group, University of Oxford, United Kingdom.
** Corresponding author.
E-mail address: samuel.albanie.academic@gmail.com (S. Albanie).
[1] Equal contribution.

https://doi.org/10.1016/j.artint.2024.104235
Received 5 November 2022; Received in revised form 11 September 2023; Accepted 12 October 2024

Available online 30 October 2024

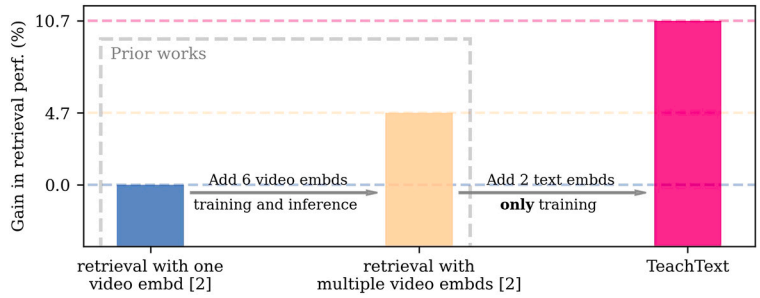


Fig. 1. Distilling the knowledge from multiple text encoders for stronger text-video retrieval. Prior work [5–7] has shown the considerable benefit of transitioning from video encoders that ingest a single modality (*left*) to multi-modal video encoders (*centre*). In this work, we show that retrieval performance can be further significantly enhanced by learning from multiple text encoders through the TeachText algorithm which imposes no additional cost during inference. Results are reported on the MSR-VTT dataset [8]. (For interpretation of the colours in the figure(s), the reader is referred to the web version of this article.)

A central theme of recently proposed retrieval methods has been the investigation of how to best use multiple video modalities to improve performance. In particular, architectures based on mixtures-of-experts [6,5] and multi-modal transformers [7] have shown the benefit of making use of diverse sets of pre-trained models for related tasks (such as image classification, action recognition and ambient sound classification) as a basis for video encoding during training and testing.

In this work, we explore whether commensurate gains could be achieved by leveraging multiple text embeddings learned on large-scale written corpora. More exactly, in the rapidly evolving field of text-to-video retrieval, the utilization of pre-trained text embeddings has emerged as a pivotal element to enhance performance. These embeddings, designed to capture linguistic nuances and semantic relationships, play a crucial role in determining how well a retrieval system matches textual queries to relevant video content. But, how do these embeddings truly influence the retrieval results, especially given their diverse origins, architectures, and training objectives? Consider a scenario where a user searches for a video depicting a 'golden retriever playing fetch at sunset.' Different text embeddings might prioritize different aspects: some may emphasize the 'golden retriever' due to the prevalent mention of the breed in its training data, while others might focus on the atmospheric quality of 'sunset' due to their architectural nuances. This raises pertinent questions: Are some embeddings more suited to certain tasks than others? How do variations in their training objectives and data influence retrieval performance? Our exploration is centred on dissecting the differential impacts of various popular pre-trained text embeddings on the retrieval tasks. Through this study, we aim to shed light on the nuances and variances each embedding introduces, thereby offering insights that will help both researchers make informed decisions in their applications. While intuitively this can easily be understood, different from video embeddings using multiple modalities and pretraining tasks, in the research community, it is less obvious that there is sufficient diversity among collections of text embeddings to achieve a meaningful boost in performance. In fact, our inspiration stems from a careful investigation of the performance of different text embeddings across a range of retrieval benchmarks (Fig. 2). Strikingly, we observe not only that there is considerable variance in performance across text embeddings, but also that *their ranking is not consistent*, strongly supporting the idea of using multiple text embeddings.

Motivated by this finding, we propose a simple algorithm, TeachText, to effectively exploit the knowledge captured by collections of text embeddings. Our approach requires a "student" model to learn from a single or multiple "teacher" retrieval models with access to different text embeddings by distilling their text-video similarity matrices into an enhanced supervisory signal. As shown in Fig. 1, TeachText is capable of delivering a significant performance gain. Moreover, this gain is complementary to that of adding more video modalities to the video encoder but importantly, unlike the addition of video modalities, does not incur additional computational cost during inference.

Moreover, our proposed method has a range of potential applications within the text-video retrieval domain. A primary concern in this area is dealing with label noise in datasets, especially those which are crowd-sourced. To address this, we highlight *denoising* as a principal application of our approach. For instance, the MSR-VTT dataset contains captions that can be ambiguous or generic, complicating the training process. Such captions, including examples like "A tutorial is presented" or "A man is writing", can be relevant to multiple videos. Using the TeachText teachers, our method offers a solution by filtering out these less precise captions, thereby improving dataset quality. This denoising strategy proves especially valuable for datasets with multiple captions per video, such as MSR-VTT and MSVD. Subsequent sections will delve deeper into the denoising application and discuss its effectiveness, particularly in conjunction with the CE+ model, showcasing the broader applicability of our proposed method.

Moreover, as we delve deeper into our research, our primary focus remains on harnessing the power of multiple text embeddings via a novel distillation approach. However, the foundational ideas of our method naturally extend to video embeddings. With this in mind, we introduce TeachVideo, a direct adaptation that integrates various video modalities. In this way, as opposed to current approaches, we use multiple video modalities only during training, while having only a small drop in performance.

Our core contributions are as follows:

1. We introduce the TeachText algorithm that makes effective use of multiple text encoders.
2. We advocate for a new method to directly learn the retrieval similarity matrix between joint query video embeddings, which is unique in the current literature. We compare this approach to existing techniques like uni-modal relationship distillation [4].

3. We apply our method to remove noise from current training datasets used in the text-video retrieval task.
4. Through experimental results, we show that our method offers notable improvements on six text-video retrieval benchmarks.

This work builds upon our previous publication at ICCV 2021 [1] in the following ways:

1. We enhance the CE+ architecture by adding two variants: one focused on minimizing parameters without sacrificing performance, and another that uses the GPT-J [2] text embeddings, leading to better performance on all benchmarks.
2. We provide additional results on the Condensed Movies Dataset (CMD) [3], highlighting models from our TeachText-based ensemble that performed well in the CMD Challenge.
3. For a deeper understanding of TeachText, we offer more detailed ablations and technical insights. An error analysis further clarifies the benefits of using multiple text embeddings.
4. We expand our denoising experiments and provide a thorough analysis of the TeachVideo approach.

2. Related work

This section synthesizes foundational and emergent literature across three pertinent domains: Video Retrieval Methods, Text Embeddings, and Knowledge Distillation/Privileged Information. We examine methods for Video Retrieval that focus on indexing and retrieving content based on textual queries. We also look into Text Embeddings, which discusses the use of linguistic representations in a range of applications, including text-to-video retrieval. Finally, we explore the concept of Knowledge Distillation, a technique for transferring knowledge between different machine learning models.

2.1. Video retrieval methods

The landscape of video content indexing and retrieval has evolved substantially over time, marked by key milestones that have expanded the boundaries of the field. Early research focused on object-specific retrieval, with pioneering work by Sivic et al. in 2003 [9]. This was followed by expansions into action-based retrieval [10] and predefined semantic categorization [11], laying the groundwork for future developments. A significant advancement came in 2014 with Self-Paced Reranking (SPaR) by Jiang et al. [12], which applied a self-paced learning framework for reranking search results. The issue of zero-shot learning then gained prominence, with Gan et al. in 2016 providing a novel approach that used external ontologies for action recognition solely based on action names [13]. They extended their zero-shot learning paradigm in 2017 to include event recognition in consumer videos, automatically selecting representative and reliable concepts for event queries using DECK (Discovering Event Composition Knowledge) from web images [14]. As the field further matured, cross-modal methods employing joint-embedding spaces rose to prominence [15,8,16–19]. These models optimized for efficient indexing by creating a shared space between video and text, while hierarchical embeddings also made a significant impact [20]. In recent times, two major themes have emerged. The first focuses on large-scale weakly supervised pretraining [21–23], leveraging audio-visual data for weak supervision. The second hones in on multimodal integration [5–7,24], which has shown remarkable gains in performance metrics [25]. Our current work engages with this latter, rapidly evolving theme of multimodal integration, adding to the ongoing dialogue in this dynamically evolving field.

2.2. Text embeddings

The field of linguistic representation through the use of learned embeddings has seen extensive exploration and analysis. Such representations, as highlighted by Mikolov et al. (2013) [26], Radford et al. (2018) [27], and Radford et al. (2019) [28], have been incorporated into a lot of natural language processing (NLP) tasks. What's noteworthy from past research is the realization that, irrespective of the vastness of pretraining, refining or finetuning these embedding structures on the specific task they are intended for can often enhance their efficacy. This perspective is supported by studies like those by Radford et al. (2018) [27] and Devlin et al. (2019) [29]. Furthermore, there's a growing consensus in the literature, underscored by Devlin et al. (2019) [29], that the more extensive the model – particularly those models that integrate multiple attention mechanisms – the more optimal its performance. Diving into a slightly different domain, Burns et al. (2019) [30] ventured into the interconnected realms of language and vision. They presented a comprehensive analysis focusing on the significance of linguistic features in vision-oriented applications. Their research also introduced a novel word embedding scheme, tailored specifically to cater to visual tasks. Drawing inspiration from these past works, our current study embarks on two primary objectives. Initially, we delve into understanding the influence of various pretrained linguistic embeddings on performance outcomes in the realm of text-to-video retrieval. Building on our findings, we subsequently introduce a novel technique that capitalizes on the synergistic benefits of integrating multiple textual embeddings.

2.3. Knowledge distillation/privileged information

The purpose of knowledge distillation is to transfer knowledge from one model (teacher) to another model (student). This idea was originally introduced in the context of decision tree simplification [33] and model compression [34], and later extended by [35] who formalised this knowledge transfer as the temperature-parameterised process of *knowledge distillation*. The concept was further generalised in the unifying framework of *generalized distillation* [36] for learning with privileged information [37] (via *similarity control* and *knowledge transfer* [38]), together with knowledge distillation [35]. Our approach distills knowledge of the similarities

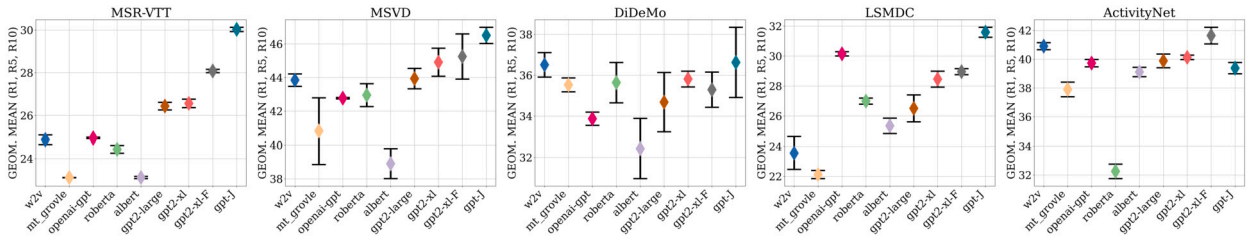


Fig. 2. The influence of the text embedding. Different text embeddings are presented along the x-axes: w2v [26], mt_grovle [30], openai-gpt [27], roberta [31], albert [32], gpt2-large [28], gpt2-xl [28], gpt2-xl-F and gpt-J [2] along with their performance in the geometric mean of $R1$-$R5$-$R10$ on five datasets. For each experiment, we report the mean (diamond) and standard deviation (error bar) of three randomly seeded runs. This study is performed using the CE retrieval architecture [6]: each model differs only in its use of pre-trained text embedding as its input. We observe a significant variance in performance when changing the text embedding, both across and within datasets. The difference in rankings across datasets suggests the presence of complementary information among different text embeddings.

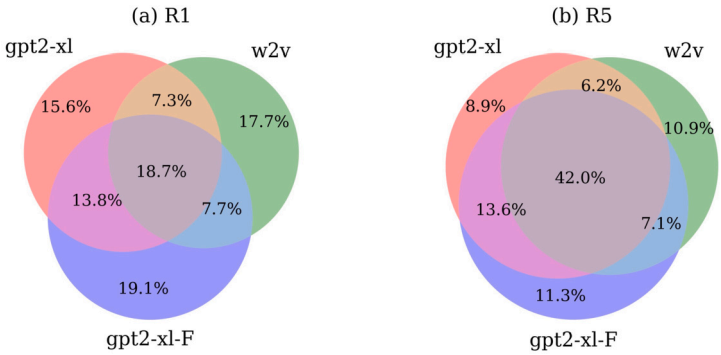


Fig. 3. Share of correctly retrieved samples based on the used pre-trained text embedding on MSR-VTT. As it can be seen, each embedding has a considerable share of sample retrieved correctly only by itself (in terms of R1 left and R5 right), further justifying our approach. Best viewed in colour.

between video and text samples into the student and therefore represents a form of generalized distillation. While most knowledge distillation methods train the student with the teacher's outputs as targets, more recent methods propose different approaches [39–41]. Knowledge distillation has consistently aimed to transfer intricate insights from a comprehensive model, often termed the "teacher", to a more compact variant known as the "student". Historically, most methodologies in this domain have guided the student model to replicate the individual output activations exhibited by the teacher for distinct data samples. One notable contribution in this context is by Park et al. [4]. Their approach, labelled relational knowledge distillation (RKD), takes a unique turn. Instead of merely focusing on individual data points, RKD emphasizes the transfer of mutual relations among data samples. By introducing distance-wise and angle-wise distillation losses, it aims to reduce relational discrepancies, rather than just mirroring the teacher's outputs. Their work, particularly in the sphere of metric learning, has even allowed student models to exceed the performance of their teachers in certain instances. However, our methodology introduces a specialized angle: we accentuate direct learning of the cross-modal similarity matrix. This nuance sets our work apart, especially when navigating the challenges in cross-modal scenarios.

3. Motivation and intuition

Recently, [28] points out that even though language representation learning systems (such as [31,32,27]) are pre-trained on vast amounts of data, they are still sensitive to slight changes in the data distribution and task specification. In this way, most systems can be viewed as *narrow experts rather than competent generalists*.

Consequently, in Fig. 2 we investigate how the usage of different off-the-shelf pre-trained text embeddings affects retrieval performance. We observe that there is significant variance both within and across datasets, suggesting that each embedding captures different types of information. Our intuition is that this information comes from the diversity of architectures, pretraining datasets and pretraining objectives, which differs across the text embeddings.

Next, we give details about the used text embeddings and summarise the key differences between them in relation to our findings. Word2vec (w2v) [26] is a lightweight text embedding that is widely used for vision tasks [42–44]. Multi-task GrOVLE (mt_grovle) [30] is an extension of w2v that is specifically designed for vision-language tasks (in our experiments, however, we find that it slightly under-performs w2v). The fine-tuned transformer language model (openai-gpt) [27] embedding is trained on a book corpus containing long stretches of contiguous text. We observe that it performs well on datasets that have longer text queries such as ActivityNet. RoBERTa [31] and ALBERT [32] are based on the BERT architecture [29] and are trained on the same data which consists of unpublished books and Wikipedia articles. RoBERTa [31] focuses on hyperparameter optimization and shows that greater model capacity leads to better performance while ALBERT [32] proposes parameter-reduction techniques to reduce memory consumption and increase training speed. In our experiments, we observe a high variation in performance when comparing the two. In contrast to the other embeddings, gpt2 [28] is trained on a crawled dataset that was designed to be as diverse as possible. We observe that

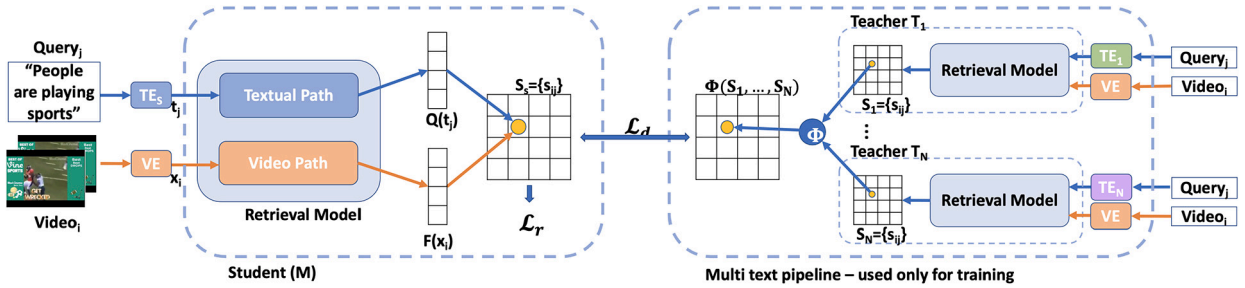


Fig. 4. TeachText teacher-student framework overview. Given a batch of input videos and queries in natural language during training, the student model, M (left) and teacher models $T_1, \ldots, T_N$ (right) each produce similarity matrices (visualised as square grids). The similarity matrix produced by M is encouraged to match the aggregated matrices of the teachers through the distillation loss $\mathcal{L}_d$ in addition to the retrieval loss $\mathcal{L}_r$. Note that both the student and teachers ingest the same video embeddings (VE), but employ different text embeddings (TE_S for the student, $\text{TE}_1, \ldots, \text{TE}_N$ for the teachers). At test time, the teacher models are discarded.

gpt2 performs most robustly in our experiments. However, it nevertheless exhibits a domain gap to each corpus (highlighted by the fact that performance increases when fine-tuning gpt2-xl, termed gpt2-xl-F throughout the paper, on descriptions from text-video retrieval datasets). Lastly, gpt-J [2] is the largest model in terms of parameters and is trained on vast amounts of textual data (825 GB).

In Fig. 3 we illustrate the number of correctly retrieved queries shared between three text embeddings on MSR-VTT: gpt2-xl, gpt2-xl-F and w2v. We observe that only approximately 19% of queries are correctly retrieved by all three of the considered text embeddings under the R1 (recall@1) metric (for recall@5, the corresponding number is 42%). The significant number of queries uniquely retrieved by each text embedding suggests the presence of complementary information across embeddings.

4. Method

Motivated by the findings from Sec. 3, our work aims to study the influence of using multiple text embeddings for text-video retrieval.

4.1. Problem description and learning setup

Let $D = \{(v_i, c_i)\}_{i=1}^{n}$ be a dataset of paired videos and captions. Following the multimodal expert approach of [5–7], for each video we assume that we have access to a collection of video embeddings (sometimes referred to as "experts") x_i extracted from the various modalities of video v_i using a pretrained video encoder (VE) in addition to a text embedding t_i (extracted using a text encoder, TE) for each caption/query c_i.[2] The objective of the text-video retrieval task is to learn a model $M(x_i, t_j)$ which assigns a high similarity value to pairings (x_i, t_j) of video and text embeddings that are in correspondence (i.e. $i = j$) and a low similarity otherwise. As is common in the literature [45,5], we parameterise the model as a dual encoder that produces joint-embeddings in a shared space such that they can be compared directly: $M(x_i, t_j) = F(x_i)^T Q(t_j) \in \mathbb{R}$ where F and Q represent the learnt video and text encoder respectively. To train the video and text encoder for the task of retrieval, we adopt a contrastive ranking loss [46]:

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

where B represents the batch size used during training, $s_{ij} = F(x_i)^T Q(t_j)$ is the similarity score between the encoded video $F(x_i)$ and query $Q(t_j)$, and m is the margin.

The key idea behind our approach is to learn a retrieval model, M, that, in addition to the loss described above, also has access to information provided by a collection of pre-trained "teacher" text-video retrieval models which are trained on the same task but ingest different text embeddings. This is described next.

4.2. TeachText algorithm

To enhance the retrieval performance of model M, we propose the TeachText algorithm which seeks to exploit cues from multiple text embeddings. An overview of our approach is provided in Fig. 4. In the initial phase of the algorithm, we train a collection of teacher models $\{T_k : k \in \{1, \ldots, N\}\}$ for the text-video retrieval task using the approach described in Sec. 4.1. The teachers share the same architecture, but each model T_k uses a different text embedding as input (extracted using a pre-trained text encoder TE_k). In the second phase the parameters of the teachers are frozen. We then proceed by sampling a batch of B pairs of videos and captions and computing a corresponding similarity matrix $S_k \in \mathbb{R}^{B \times B}$ for each teacher T_k (Fig. 4 right). These N similarity matrices are then

[2] These embeddings are produced by models that have been trained on relevant tasks (such as action recognition for the video encoder and language modelling for the text encoder.).

combined with an aggregation function, $\Phi : \mathbb{R}^{N\times B\times B} \rightarrow \mathbb{R}^{B\times B}$, to form a single supervisory similarity matrix (Fig. 4, centre-right). Concurrently, the batch of videos and captions are likewise processed by the student model, M, which produces a further similarity matrix, $S_s \in \mathbb{R}^{B\times B}$. Finally, in addition to the standard retrieval loss (Eq. (1)), a distillation loss, $\mathcal{L}_d$, encourages S_s to lie close to the aggregate $\Phi(S_1, \ldots, S_N)$. The algorithm is summarized in Algorithm 1. During inference, the teacher models are discarded and the student model M requires only a single text embedding. Next, we give details of the distillation loss used to drive the similarity matrix learning.

Algorithm 1 TEACHTEXT algorithm.

Phase 1: *Learn teacher models*
  Train N teacher models $T_k = (F_k, Q_k)$, $k \in \{1, \ldots, N\}$ using the training pairs (x_i, t_j^k) where t_j^k represents the text modality used by teacher T_k in a standard retrieval training setup (Sec. 4.1).
Phase 2: *Learn the student model,* $M = (F, Q)$
  for minibatch of B paired samples $\{(v_i, c_i)\}$ **do**
    For each pair (v_i, c_i) extract video experts and text embedding pairs (x_i, t_i) using VE and TE_S.
    Compute student similarity matrix $S_s = [s_{ij}]$ where $s_{ij} = F(x_i)^T Q(t_j)$ for $i, j \in \{1, \ldots, B\}$
    Compute the loss $\mathcal{L}_r$ via Eqn. (1) using S_s.
    for teacher T_k, $k = 1, \ldots, N$ **do**
      For each pair (v_i, c_i) extract the video experts and text embedding pairs (x_i, t_i^k) using VE and TE_k.
      Compute the similarity matrix $S_k = [s_{ij}^k]$ where $s_{ij}^k = F_k(x_i)^T Q_k(t_i^k)$ for $i, j \in \{1, \ldots, B\}$
    end for
    Compute aggregated teacher matrix $\Phi(S_1, \ldots, S_N)$.
    Compute the loss $\mathcal{L}_d$ between S_s and $\Phi(S_1, \ldots, S_N)$ via Eqn. (2).
    Update M with gradients computed from the composite loss $\mathcal{L} = \mathcal{L}_r + \mathcal{L}_d$.
  end for

4.3. Learning the similarity matrix

As noted in Sec. 4.1, the essence of the retrieval task is to create a model that is able to establish cross-modal correspondences between video and text/queries, assigning a high similarity value to a pairing in which a query accurately describes a video, and a low similarity otherwise. This renders the similarity matrix a rich source of information about the knowledge held by the model. In order to be able to transfer knowledge from the teachers to the student, we encourage the student to produce a similarity matrix that matches an aggregate of those produced by the teachers. In this way, we convey information about the text and video correspondences without strictly forcing the student to produce the same embeddings as the teachers. To this end, we define the similarity matrix distillation loss as:

$$\mathcal{L}_d = \frac{1}{B} \sum_{i=1}^{B} \sum_{j=1}^{B} \left[l((\Phi(S_1, \ldots, S_N))_{ij}, s_{ij}) \right], \tag{2}$$

where B represents the batch size, N the number of teachers, $S_k, k = 1 \ldots N$ the similarity matrix produced by teacher k, $\Phi(S_1, \ldots, S_N)$ represents the aggregation function of the teacher similarity matrices and s_{ij} represents the student similarity between the samples i and j. Finally, inspired by prior distillation work [4], we implement l as a Huber loss defined as follows:

$$l(x, y) = \begin{cases} \frac{1}{2}(x-y)^2 & \text{if } |x-y| \leq 1, \\ |x-y| - \frac{1}{2} & otherwise \end{cases}. \tag{3}$$

We explored several forms of aggregation function and found that a simple element-wise mean, $\Phi(S_1, \ldots, S_N) = \frac{1}{N} \sum_{k=1}^{N} S_k$, worked well in practice.

The idea of learning directly the cross-modal similarity matrix is, to the best of our knowledge, novel. It draws inspiration from the work of relational knowledge distillation [4] which considered the idea of learning from relationships and introduced two algorithms to implement this concept in a uni-modal setting through pairwise and triplet distance sampling. We compare our matrix learning approach with theirs in Sec. 5.

4.4. Student model

A key advantage of our approach is that it is agnostic to the architectural form of the student and teachers, and thus the student (and teachers) can employ any method from the current literature. We test our TEACHTEXT algorithm using three different recent works MoEE [5], CE [6], MMT [7] as the student and teacher base architectures. All these works employ multimodal video encoders for the text-video retrieval task. For more details, please consult the original paper of each method.

Establishing a stronger baseline: CE+, CE-L and CE-J. In our exploration beyond the initial models, we studied the efficacy of our approach on an enhanced model. This model is built upon the CE architecture from [6], but incorporates several technical improvements. Our goal was to create a stronger baseline for evaluating the TEACHTEXT algorithm. Our refinements began with the input embedding selection. From our investigations, we found that certain video modalities, specifically face and OCR as used by [6], did not consistently offer performance benefits. As a result, we decided to exclude them from the video encoder's inputs. Additionally, we revisited the

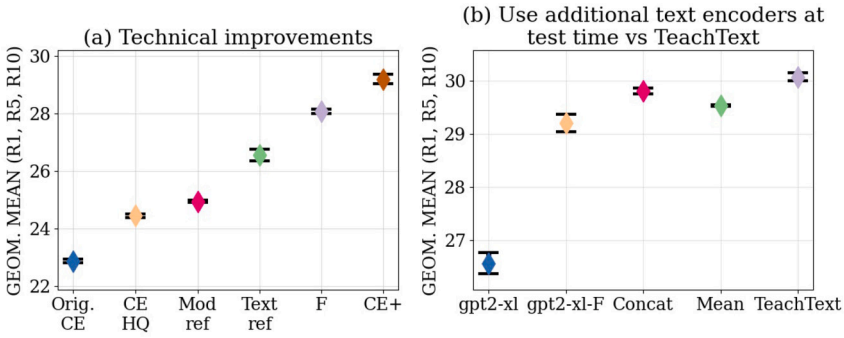


Fig. 5. (a) Baseline improvements. The y-axes (scaled for clarity) denote retrieval performance on MSR-VTT. We begin by presenting the performance of the original CE model [6]. Firstly, we correct compression artefacts in the pre-processing used for embedding extraction (*CE HQ*). Secondly, we refine the used video modalities and text modalities (*Mod ref* and *Text ref*). Finally, we finetune the text embedding (*F*) and change the optimizer to Adam [51], thus obtaining the *CE+* baseline. **(b) Use of additional text embeddings at inference time**. All experiments were performed with the same architecture [6], but with different text embeddings: gpt2-xl (first bullet), gpt2-xl-F (second bullet), the concatenation of gpt2-xl and gpt2-xl-F (third bullet), the mean of gpt2-xl and gpt2-xl-F (fourth bullet) and using TeachText (last bullet). By using multiple text embeddings at test time, which introduces an overhead, a boost in performance is obtained. However, by using TeachText there is no additional overhead at test time and the performance is superior.

expert selection outlined by [6]. Instead of using the *Action(IG)* from the I3D model [47], we opted for an R2P1D model [48]. This model, which aligns with the *Action(IG)* architecture, was first trained on IG-65m [49] and later fine-tuned on the Kinetics dataset [47]. Further enhancing the model, we integrated the gpt2-xl text embedding from [28]. Following insights from [7], we fine-tuned this embedding using captions from our target dataset, leading to noticeable improvements. One of our main contributions is the introduction of the CE+ model. It includes all the changes we've discussed, and its performance is studied in Sec. 5.5 and Fig. 5a. This CE+ model serves as an advanced baseline in our study. We also introduce two additional variants: CE-L and CE-J. CE-L aims to be efficient with fewer trainable parameters but still delivers competitive performance, using *w2v* for text embedding. In contrast, CE-J is designed for superior performance and employs the robust *gpt-j* text embedding. In summary, our work spans several base architectures for the student model, referencing works like [5–7], and importantly introduces the CE+, CE-L, and CE-J models.

4.5. Teacher models

The teacher models use the same architecture as the student model. Concretely, for each of the four base architectures described in Sec. 4.4, we create a pool of multiple teachers, each using a different pretrained text embedding as input. The candidate text embeddings we consider in this work are mt_grovle [30], openai-gpt [27], gpt2-large [28], gpt2-xl [28], w2v [26]. In this way, we obtain a set of up to five models that form the teachers T_k, $k = 1..5$ used by TeachText. In addition to results reported in our conference paper [1], we extend the embedding studies to incorporate gpt-J [2].

4.6. Training and implementation details

To effectively train our final student model, we synergistically combine the retrieval loss with the proposed distillation loss, represented as $\mathcal{L} = \mathcal{L}_r + \mathcal{L}_d$. Our CE+ models have been trained using Pytorch [50], leveraging the capabilities of the Adam optimizer [51]. We've set the learning rate at 0.001 and applied a weight decay of 1E-5. When an alternative base architecture, other than our proposed CE+, is utilized, we ensure the adoption of the same hyper-parameters specified in the public codebase for the respective method, whether it's CE[3] or MMT.[4] For the MoEE model, we rely on the reimplementation provided by the original authors of the CE method [6]. Importantly, TeachText does not introduce any new trainable parameters or modalities to the final model. When employing TeachText for training, only the added loss term $\mathcal{L}_d$ is integrated, while all other hyper-parameters are retained without modification. TeachText has no inference overhead. However, during training, multiple teacher models are required with a total maximum training time varying from 45 minutes up to 14h depending on the dataset.

5. Experimental setup

5.1. Datasets description

To provide an extensive comparison we test our approach on seven video datasets that have been explored in recent works as benchmarks for the task of text-video retrieval. We follow the same experimental setup as prior works [6,7,20,24]. Next, we give details about all the datasets used.

[3] https://github.com/albanie/collaborative-experts.
[4] https://github.com/gabeur/mmt.

MSRVTT [52] contains 10k videos, each having 20 captions. In order to test the retrieval performance, we report results on the official split which contains 2990 videos for the test split and 497 for validation, following the setup used in [6]. We perform most of our ablations on this split. To enable comparison with as many other methods as possible, we also report results on the 1k-A split as used in [6,7,24]. For this split, we report the performance after training 100 epochs. The split contains 1000 video candidates for testing and 9000 for training. We use the same candidates as defined in [6] which are used by other works [18,7,24].
MSVD [53] contains 80k English descriptions for a total of 1970 videos. We use the standard split of 1200 (training), 100 (validation) and 670 (testing) as used in other works [54,8,6]. The videos from MSVD do not have audio streams.
DiDeMo [55] contains 10464 videos sourced from a large-scale creative commons collection [56] and features moments of unedited, diverse content (concerts, sports, pets etc.). The dataset comprises 3-5 pairs of descriptions per video. We adopt the paragraph-video retrieval protocols used by [57,6] and use splits corresponding to 8392 train, 1065 validation and 1004 test videos.
LSMDC [58] contains 118081 short video clips extracted from 202 movies. Each clip is described by a caption that is either extracted from the movie script or from transcribed DVS (descriptive video services) for the visually impaired. There are 7408 clips in the validation set and the testing is performed on 1000 videos from movies that are disjoint from the training and val sets as described in the Large Scale Movie Description Challenge (LSMDC).[5]
ActivityNet [59] contains 20k videos extracted from YouTube and has around 100K descriptive sentences. We follow the same paragraph-video retrieval setup as used in prior works [57,6] and report results on the `val1` split: we use 10009 videos for training and 4917 videos for testing.
VaTeX [60] contains 34911 videos with multilingual captions (Chinese and English). There are 10 captions per video for each language. We follow the same protocol as in [20,24] and split the validation set equally (1500 validation and 1500 testing videos). In this work, we only use the English annotations.
QuerYD [61] contains 1815 videos in the training split and 388 and 390 for validation and testing. The videos are sourced from YouTube and cover a diverse range of visual content. The dataset contains 31441 descriptions, from which 13019 are precisely localized in the video content (with start time and end time annotations) and the other 18422 are coarsely localized. For this work, we do not use the localization annotations and report results for the official splits.
Condensed Movies [3] is a large scale dataset containing clips from movies annotated with detailed captions. It has around 30k captioned video clips extracted from around 3k movies. In order to enable future comparisons, we report results on the validation split (2200 clips) used for the CMD-Challenge[6] while we use the training split (31k clips) for training.

5.2. Metrics

To assess performance, we follow prior work (e.g. [16,5,17,18,21,6,7]) and report standard retrieval metrics. These include R@K (recall at rank K, where higher is better) and MdR (median rank where lower is better). To maintain concision, for certain analyses we report the geometric mean of R@1, R@5, and R@10 rather than individual metrics (this statistic aims to be representative of overall retrieval performance). The numbers are reported for the task of retrieving a video given text queries `t2v` as well as for the opposite task `v2t`. For each experiment, we report the mean and standard deviation of three randomly seeded runs.

5.3. Video embeddings (experts) description

In this work, we used the set of pretrained experts considered by the authors of [6]. For completeness, we summarise below the manner in which these experts were extracted.
Two form of action experts are used: *Action(KN)* and *Action(IG)*. The former is an I3D architecture trained on Kinetics [47], which produces 1024-dimensional embeddings from frame clips extracted at 25fps and centre cropped to 224 pixels. The *Action(IG)* model is a 34-layer R(2+1)D model [48] that has ben trained on IG-65m [49]: it operates on frames extracted at 30 fps in clips of 8 at 112 $\times$ 112 pixel resolution.
Two forms of object experts are used, named *Obj(IN)* and *Obj(IG)*. They are produced from frame-level embeddings extracted at 25fps. The *Obj(IN)* model consists of an SENet-154 backbone [62] which has been trained on ImageNet for image classification. *Obj(IG)* is formed from a ResNext-101 [63] extractor which was trained on weakly labelled Instagram data [64]. For both models, frames are resized to 224×224 pixels.
The audio expert is produced using the VGGish model [65], trained for audio classification on the YouTube-8m dataset.
The scene expert is an embedding that is extracted at 25 fps from a centre crop of 224×224 pixels. The model, which is pretrained on Places365 [66], uses a DenseNet-161 [67] architecture.
The speech expert is produced using the Google Cloud API (to transcribe the speech content).

5.4. Text embeddings description

We use several text embeddings as follows:

[5] https://sites.google.com/site/describingmovies/previous-years/lsmdc-2019.
[6] https://www.robots.ox.ac.uk/~vgg/research/condensed-movies/challenge.html.

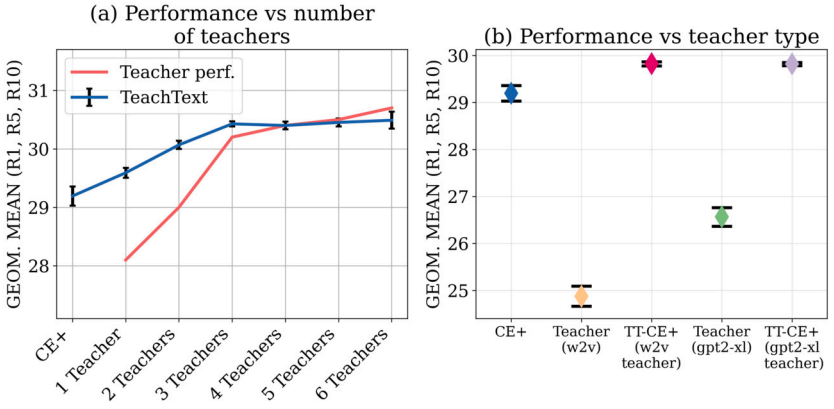


Fig. 6. (a) Teacher study. We show the influence of learning from different number of teachers on the MSR-VTT dataset (all students share the same CE+ model, y-axes scaled for clarity). The teachers were added in the following order: gpt2-xl, w2v, gpt2-xl-F, mt_grovle, openai-gpt, gpt2-large. The performance of the combined teachers grows as more teachers are added, however it reaches a plateau after the first 3 teachers. The trend is similar for student performance. **(b) Performance vs teacher type.** We study the influence of teachers with different text embeddings at input: w2v and gpt2-xl. The first point represents the performance of the student without using TeachText. We observe a boost in performance independent of the nature of the teacher.

Table 1
Method generality. Retrieval performance on various datasets when applying TeachText on top of different base models. Instances in which TeachText brings an improvement over the base architecture are highlighted in bold. We observe that TeachText yields a consistent boost in performance regardless of base architecture.

Model	MSRVTT		MSRVTT 1k-A		MSVD		DiDeMo		LSMDC		ActivityNet	
	Base	TeachText	Base	TeachText	Base	TeachText	Base	TeachText	Base	TeachText	Base	TeachText
MoEE	$24.4_{\pm 0.1}$	$\mathbf{25.8}_{\pm 0.1}$	$41.6_{\pm 0.4}$	$\mathbf{43.4}_{\pm 0.6}$	$41.8_{\pm 0.3}$	$\mathbf{43.2}_{\pm 0.5}$	$33.2_{\pm 1.4}$	$\mathbf{40.2}_{\pm 0.7}$	$23.8_{\pm 0.4}$	$\mathbf{26.0}_{\pm 0.5}$	$40.1_{\pm 0.3}$	$\mathbf{45.2}_{\pm 0.1}$
CE	$24.4_{\pm 0.1}$	$\mathbf{25.9}_{\pm 0.1}$	$42.0_{\pm 0.8}$	$\mathbf{43.8}_{\pm 0.3}$	$42.3_{\pm 0.6}$	$\mathbf{42.6}_{\pm 0.4}$	$34.2_{\pm 0.4}$	$\mathbf{39.5}_{\pm 0.5}$	$23.7_{\pm 0.3}$	$\mathbf{25.5}_{\pm 0.5}$	$40.4_{\pm 0.3}$	$\mathbf{45.0}_{\pm 0.6}$
MMT	-	-	$44.7_{\pm 0.4}$	$\mathbf{45.6}_{\pm 0.7}$	-	-	-	-	$24.6_{\pm 0.7}$	$\mathbf{25.9}_{\pm 0.6}$	$44.0_{\pm 0.4}$	$\mathbf{47.9}_{\pm 0.4}$
CE+	$29.2_{\pm 0.2}$	$\mathbf{30.4}_{\pm 0.0}$	$50.3_{\pm 0.2}$	$\mathbf{50.9}_{\pm 0.4}$	$46.5_{\pm 1.0}$	$\mathbf{46.6}_{\pm 0.5}$	$35.8_{\pm 0.4}$	$\mathbf{40.4}_{\pm 0.4}$	$28.1_{\pm 0.3}$	$\mathbf{30.7}_{\pm 0.3}$	$39.7_{\pm 0.0}$	$\mathbf{46.3}_{\pm 0.2}$
CE-L	$25.5_{\pm 0.1}$	$\mathbf{26.9}_{\pm 0.1}$	$45.7_{\pm 0.2}$	$\mathbf{46.5}_{\pm 0.8}$	$41.3_{\pm 0.5}$	$\mathbf{42.6}_{\pm 0.7}$	$36.4_{\pm 0.5}$	$\mathbf{41.5}_{\pm 0.4}$	$24.1_{\pm 0.2}$	$\mathbf{25.9}_{\pm 0.3}$	$39.6_{\pm 0.5}$	$\mathbf{45.7}_{\pm 0.2}$

mt_grovle [30] is a "vision-sensitive" language embedding which is adapted from w2v using WordNet and an original vision-language graph built from Visual Genome [68]. The size of the final pre-trained embedding is 300.
OpenAI-GPT [27] is a pre-trained text embedding which uses transformers [69] and language modelling on a large corpus (the Toronto Book Corpus) (the final model has 110M params). The size of the final pre-trained embedding is 768.
RoBERTa [31] is a BERT-based embedding [29]. The model is trained longer with bigger batch size on more data, having 125M params. The size of the final pre-trained embedding is 768.
ALBERT [32] is a lightweight modification to BERT [29] which overcomes some memory limitations, having 11M params. The size of the final pre-trained embedding is 768.
GPT2-large [28] is a transformer-based [69] model trained on even more data (40 Gb of text) without any supervision, having 774M params. The size of the final pre-trained embedding is 1280.
GPT2-xl [28] is similar to GPT2-large, but has more parameters (1558M params). The size of the final pre-trained embedding is 1600.
W2V [26] is one of the most popular text embeddings used in vision tasks. It uses a neural network model to learn word representations. The size of the final pre-trained embedding is 300.
GPT-J [2] is a new powerful text embedding released recently. It has around 6B learnable parameters and is trained on 825Gb of textual data. The size of the final pre-trained embedding is 50400 and we use PCA to reduce the dimensionality to 2048.

5.5. Ablations

In this section we present an extensive study of our proposed approach. Following the setup used in prior works [6,7] we conduct ablations on the MSR-VTT dataset [52], except where otherwise stated.
Baseline improvements. We propose CE+ as an additional baseline which consists of a series of technical improvements to the model of [6]. As seen in Fig. 5a each modification described in Sec. 4.4 brings additional gain over the base architecture. We observe in particular that finetuning the text embedding on the target dataset has a high influence, further highlighting the critical role played by text embeddings and justifying their study. In addition to other changes we found that certain video embedding expert features were highly sensitive to jpeg compression choices used in video pre-processing. To address this, we re-extracted features from video frames densely extracted with minimal jpeg compression (corresponding to the use of ffmpeg [70] and the `-qscale:v 2` flag). In order to be fair in our comparisons, we apply this correction everywhere. For a fair comparison, we re-train MoEE [5] and CE [6]. In Sec. 5.11 we report the results of re-training the methods [5,6] using these embeddings extracted with the updated pre-processing which yields a higher performance than reported in the original papers.

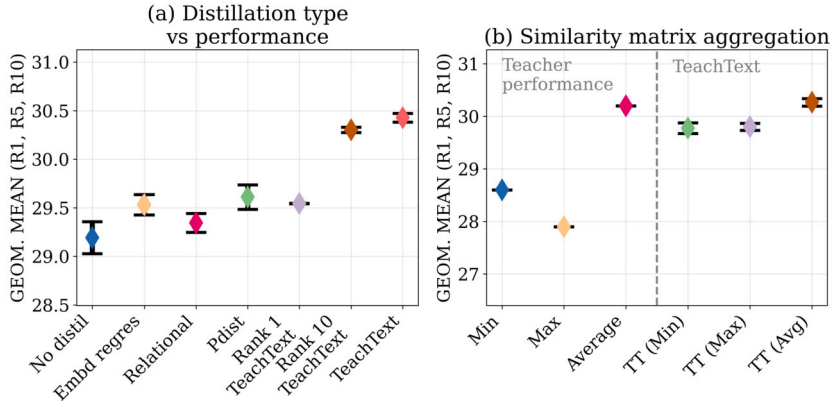


Fig. 7. (b) Distillation type. Presenting various alternatives for distilling the information from the teacher: *Embd regress* which is a classical approach where the query and video joint embeddings are directly regressed based on the embeddings given by the teacher, *Relational* distillation [4] which preserves intra-text and intra-video relationships, pairwise distance distillation (*Pdist* - adapting [4] for cross modal relationships), ranking distillation inspired by [71] at Rank 1 and Rank 10 and TeachText. The first bullet represents the student without distillation. **(b) Similarity matrix aggregation.** We present a comparison of different similarity matrix aggregation: *min*, *max* and *average*. As can be seen, the average aggregation has the best results (both when evaluating the teacher standalone or in conjunction with our TeachText algorithm).

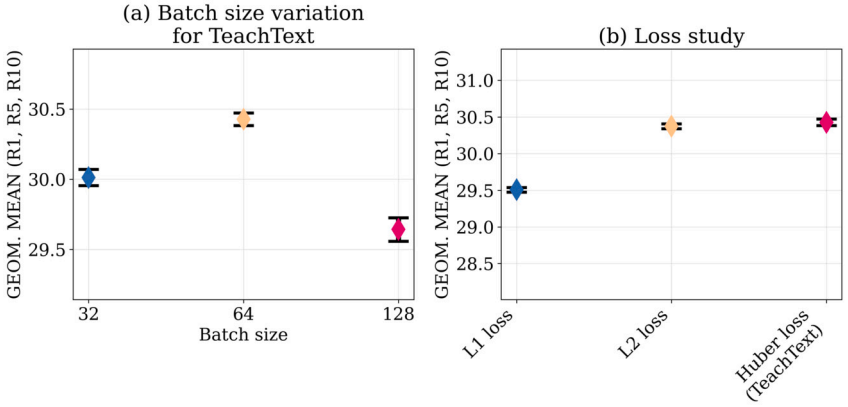


Fig. 8. (a) Batch size variation. We vary the batch size for the MSR-VTT dataset to see how this affects the performance. We observe that batch size influences performance. The underlying architecture used for this experiment is CE+. **(b) Loss study.** In this picture, we show how various distillation losses (L1, L2, Huber) affect the performance.

Additionally for the DiDeMo [55] dataset we observed that prior work [6] uses the whole clip to pre-extract the features while the captions describe only the first 30 seconds as described in [55]. We address this by employing only the pre-extracted features corresponding to the first 30 seconds of video.

Using multiple text embeddings during inference. TeachText makes no use of additional information at test time. However, it is natural to ask whether the additional text embeddings can be trivially included as part of the model architecture. In Fig. 5(b) we compare our approach with some relatively simple text embedding aggregation techniques, which require access to multiple text embeddings during both training and inference. We observe that TeachText outperforms these aggregation techniques such as direct concatenation or mean of the text embeddings, suggesting that the proposed method is effective in capturing the additional information given by multiple text embeddings. Moreover, the text encoder of existing systems [5–7] typically employs many parameters, so adding multiple text embeddings to the architecture adds a significant number of parameters (100M+). For example, the concatenation of two text embeddings (provided that they have the same size) almost doubles the total number of parameters for CE+ (approx 240M learnable parameters, yielding total model sizes of 503.98M vs 262.73M for CE+). In contrast, when employing TeachText, no parameters are added.

Teacher variation. The teacher models share the same architecture with the student, but use a different text embedding. We next conduct an ablation on the influence of the number of used teachers. We observe in Fig. 6a that performance increases with the addition of more teachers. However, since the combined performance of the teachers after adding more than 3 remains similar, we do not obtain further improvement. Thus, for our final experiments presented in Sec. 5.11 we use a combination of three teachers. The text embeddings corresponding to these three teachers are: w2v [26], gpt2-xl [28] and gpt2-xl-F (gpt2-xl finetuned on the captions from the target dataset). Moreover, in Fig. 6b, we study how changing the embedding used by the teacher model affects the final performance. We observe that even though the model ingesting w2v embeddings has a significantly lower performance than the student model without using TeachText, there is a significant gain when learning from the teacher which uses w2v. This again

indicates that there is additional information captured by using a different text embedding which can be exploited by TeachText.

Distillation ablation. We compare the proposed learning of the similarity matrix with other distillation alternatives. As seen in Fig. 7a, our proposed approach is effective in capturing the relationships between video and text. Firstly, we test our approach against a more classical distillation setup, that does not follow the idea of relational distillation, where we directly regress the embeddings given by the teacher (Embd regress). Next, we provide comparisons between TeachText and several other possible instantiations of relational distillation [4]. Indeed, given the highly general nature of [4], TeachText can be interpreted within this framework as a particular relational configuration that employs cross-modal distillation through batches of similarity matrices. Since the original work of [4] considered single-modality applications, we explore two variations of [4] as baselines for the text-video retrieval task. The first one (Relational), preserves the same intra-text and intra-video relationships independently. We use the same cost function as in [4] and enforce it on both video and text embeddings. The second approach (Pdist), uses the cross modal pairwise distances as a relation measure between text and video as opposed to the similarity matrix. While these methods indeed bring a gain, we observe that TeachText is most effective.

We also provide a baseline inspired by the work of [71] that highlighted the benefits of considering only the top K predictions given by the teacher. To do so, we match the similarities supplied by TeachText only for the top K ranks given by the teacher rather than for the whole mini-batch. We show the performance for K = 1 and K = 10 (*Rank 1* and *Rank 10* presented in Fig. 7b). Restricting to only the top K predictions when distilling the similarity matrix results in a slight drop in performance.

Method generality. To demonstrate the generality of TeachText, we test it on three strong methods [5–7] in addition to the proposed CE+ baseline. In Table 1 we observe a consistent gain in performance, independent of the base architecture. Moreover, a gain is achieved across all datasets tested, having over 5% absolute gain on DiDeMo and ActivityNet datasets for MoEE, CE and CE+ models. Note that for MMT [7] we report results on the datasets included in the public implementation provided by the authors.[7] We also introduce a new CE-L base architecture. This is similar to the CE [6] and CE+, but uses `w2v` as the text embedding, greatly reducing the number of architectural parameters. In Table 1, we observe that our method TeachText is effective even when using this lightweight architecture. This architecture also has the lowest number of parameters when compared to prior works as can be seen in Table 6, 7, 8, 9, 10, 11, 12, 13.

Similarity matrix aggregation study. In Fig. 7b we present several similarity matrix aggregation strategies: *min*, *max* and *average*. We observe that using the mean of the similarity matrices is most effective. Because of this, we use the mean as the final aggregation technique in our TeachText algorithm.

Batch size variation. In Fig. 8a we vary the batch size during training on the MSR-VTT dataset to investigate its influence on performance. We observe that we obtain the best result using the same batch size as for the method without applying the TeachText algorithm (64 in this case).

Loss study. In addition to the Huber loss (also employed by [4]), we compare against L1 and L2 losses. As observed in Fig. 8b, the Huber loss performs better than L1 loss and slightly better than L2.

Quantity of training data vs performance. We next study how training data quantity influences the proposed method. In Fig. 9a we observe that by using the TeachText with increasing quantities of data, the performance gap increases, suggesting that its benefit may prove to be more useful in larger scale dataset scenarios.

Distillation loss weight vs performance. Our final loss $\mathcal{L}$ is the sum of the distillation loss $\mathcal{L}_d$ and the retrieval loss $\mathcal{L}_r$ as described in Algorithm 1. In this ablation study, we investigate the influence of varying the weight of each loss. We explore variants of the distillation loss weight as follows $\mathcal{L} = \alpha * \mathcal{L}_d + (1 - \alpha) * \mathcal{L}_r$, for different values of $\alpha \in [0.1, 1]$. In Fig. 9b, we observe the best results if the two losses have equal weight.

Mixture of architectures. In all experiments so far, the only model difference between the teacher and the student has been the pre-trained text embedding fed to the model. However, our method is not limited to this constraint. In this ablation, we investigate the influence of using teachers with different underlying architectures. Our preliminary results shown in Fig. 10a indicate that using a mixture of architectures as teachers brings limited gains. This suggests that, in contrast to using different text embeddings, architectural changes introduce limited diversity into the teacher mixture.

Joint embedding size variation. As presented in Sec. 4, the student model learns a joint embedding between text and video. In this experiment, we vary the dimensionality of this embedding. In Fig. 10b, we observe that the embedding size has a slight influence on performance. The best results are obtained using an embedding size of 768 which we use in all the experiments.

Computational cost While TeachText does not incur any computational cost during inference, for training we need to train multiple teacher models. Depending on the dataset, the training time of a model varies between 15 minutes to 4.5 hours on one Nvidia P40 GPU. In Table 2 we present an analysis with the required training time along with the number of parameters for the teacher models. For the final model TT-CE+ we are using three teachers, so the total training time of the teacher varies between 45 minutes to 14h, depending on the dataset. Please note that the teachers can be trained in parallel.

5.6. Increasing text embedding capacity

In this experiment, we investigate the use of the recently released GPT-J [2] model, a powerful text embedding trained on large volumes of text data. This model uses a particularly large embedding dimensionality (50,400), so we employ PCA on every dataset,

[7] https://github.com/gabeur/mmt.

Table 2
Training time. Analysis for the training time for the teachers based on the dataset.

Model	Dataset	Params	Avg Teacher training time
TT-CE+	MSR-VTT	262M	3 x 55.3m
TT-CE+	MSVD	87M	3 x 71.8m
TT-CE+	DiDeMo	99M	3 x 15.4m
TT-CE+	LSMDC	388M	3 x 80.5m
TT-CE+	Activity-Net	376M	3 x 271.8m

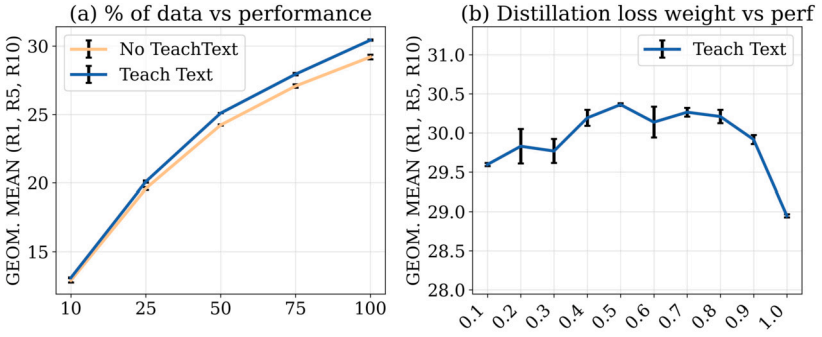


Fig. 9. (a) Amount of training data vs performance. We observe that the rate of improvement induced by TEACHTEXT increases as training data increases. **(b) Distillation loss weight vs performance.** We obtain the best performance if the distillation loss $\mathcal{L}_d$ and the retrieval loss $\mathcal{L}_r$ have the same weight.

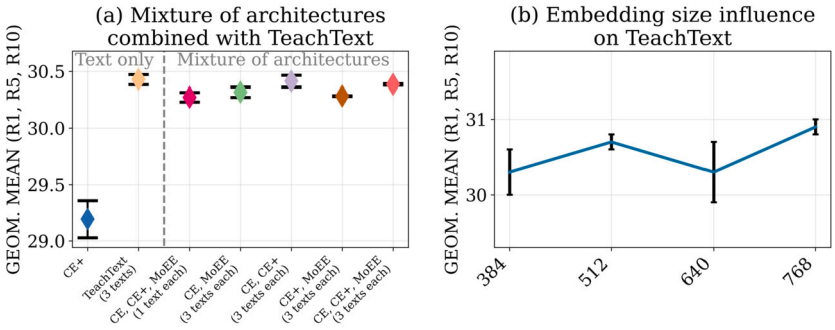


Fig. 10. (a) Mixture of architectures. We perform preliminary experiments to explore whether our method benefits from learning from teachers that do not share the same architecture. The x axis corresponds to the models which are used as teachers. In cases labelled with *3 text each*, we used three different variations of each architecture as teachers, accounting for a total number of *no. methods * 3* teachers. We observe limited benefit in using multiple architectures as teachers. **(b) Joint-Embedding size variation.** We vary the size of the underlying joint embedding used by the student model and apply TEACHTEXT it. We obtain the best results using a 768 dimensional embedding.

using the training split, to reduce its dimensionality to 2,048. We observe that GPT-J yields performance gains, as shown in Table 6, 7, 8, 9, 10, 11, 12, 13 (note that the model termed TT-CE-J uses GPT-J as text embedding).

5.7. Method application - denoising

One immediate application of our method is data denoising. Existing real-world text-video datasets for the retrieval task suffer from label noise which can harm training. More concretely, in crowd-sourced datasets such as MSR-VTT there are some captions that are highly ambiguous/generic (e.g. “A tutorial is presented”, “Clip showing different colours”, “A man is writing”) and can describe multiple videos from the dataset. We therefore propose to use TEACHTEXT teachers to filter out such cases. For this scenario, we simply remove low-ranked predictions given by teachers and retrain the student using only the new samples. Specifically, we remove all sentences for which the correct video is not ranked in the top 40 results from the training set. This method is best-suited for datasets where multiple captions per video are available, ensuring that we can remove noisy captions without removing the video itself from training. We apply the denoising method on the MSR-VTT and MSVD datasets (which have multiple captions per video) with the CE+ model. As seen in Fig. 11a, this can be an effective way of further improving the results. This process can be applied to either the student or the teacher and in each case it brings an improvement as shown in Fig. 11a. Note that denoising is not used in other ablations unless otherwise specified.
In Fig. 11b we vary the threshold used to filter out sentences from the training set, where we observe that denoising brings a boost in performance. We also observe that the best filtering threshold for MSR-VTT is rank 40. This method turns out to be effective in

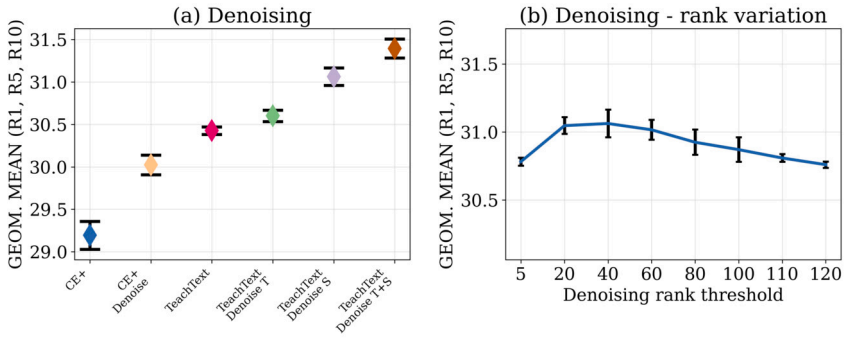


Fig. 11. (a) Denoising. We observed in preliminary studies that some of the captions available in MSR-VTT and MSVD are noisy and may actively harm the training process. Here, we present the effect of denoising on retrieval performance on MSR-VTT (y-axes scaled for clarity). We denote by *S* the cases in which the student was trained using denoising and by *T* the cases in which the teachers were trained using denoising. We estimate the degree of noise present in a caption by looking at the teacher rank and drop the caption if necessary. We observe the effectiveness of denoising when applied in isolation (*CE+* vs *CE+ Denoise*) and in conjunction with the full TeachText method. Moreover, we observe that the method increases in efficiency if we denoise both the student and the teacher (*TeachText Denoise T+S)*. **(b) Rank variation for denoising.** The denoising involves dropping captions that are assigned a low ranking by the teacher for the training set. In this experiment, we vary the rank below which we drop sentences. Please note that for a rank of 5 (on the training set) the amount of dropped sentences is approximately 46%. Note that MSR-VTT has 20 captions per video, so after applying this filter we keep on average 10 captions per video.

Table 3
Denoising - kept vs removed queries comparison. In order to see if there are any differences, we compute the length and the specificity score [72] for the sentences kept and removed in the denoising process. As it can be seen, the removed queries are shorter and less specific.

Average	Kept queries	Removed queries
Length	9.39	6.98
Specificity	0.37	0.32

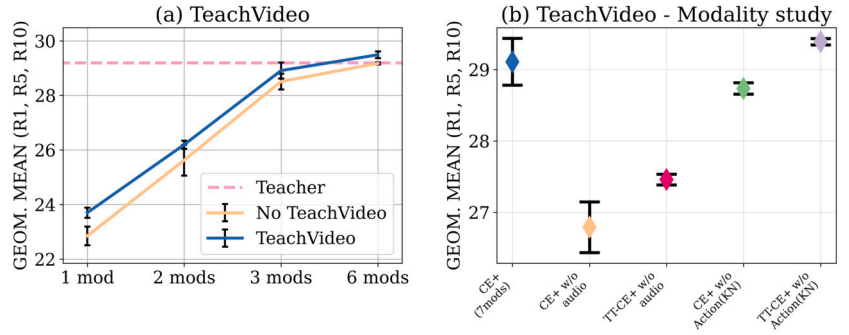


Fig. 12. (a) TeachVideo - Extension to video side modalities. We observe that our method can be effective in taking advantage of the additional information brought by using multiple video side modalities, without incurring computational overhead at test time. **(b) TeachVideo- Modality study.** In this figure we study if the nature of the removed modality affects the performance. We observe that if we remove the *audio* modality from the student there is a significant loss in performance and, while TeachVideo brings an improvement, the *audio* modality cannot be distilled entirely. On the other hand, by employing TeachVideo we can remove the *Action(KN)* from the student without observing a drop in performance.

reducing noise for retrieval datasets. Our final results when comparing with prior methods are presented using denoising on MSR-VTT and MSVD datasets. For MSVD dataset we use rank 100 as a filtering threshold.

5.8. *TeachVideo - extension to video modalities*

While the focus of this work is the use of multiple text embeddings, it is natural to consider whether this approach can be extended to video encoder modalities. Thus, we introduce the TeachVideo algorithm which follows the same setup as the original TeachText, but now the teacher has access to multiple video modalities instead of multiple text modalities. In this study, all students and all teachers use the same text embedding to assess the gains due to TeachVideo. By employing TeachVideo we retain the computational advantage of requiring fewer video modalities during inference. The experiments presented in Fig. 12a suggest that TeachVideo can bring a boost over the original student. We believe this extension may be useful in scenarios in which limited computational resources are available during inference. We also study whether the selection of specific types of modalities affects

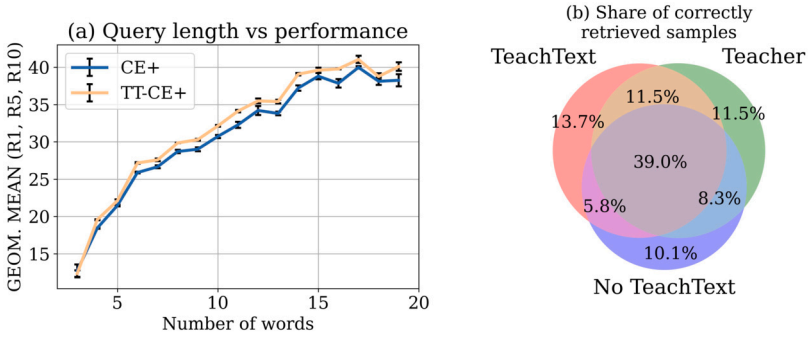


Fig. 13. (a) Performance vs query length. We present the performance vs query length for the MSR-VTT dataset. As it can be seen, there is a clear correlation between the performance and the query length.**(b) Share of correctly retrieved samples in terms of R1 when using TeachText on the MSR-VTT test set.** We show the case where we learn from 3 teachers. We can see that the model with TeachText, preserves most of the knowledge from the student without TeachText, but also acquires new information from the teacher (yellow area). Best viewed in colour.

Table 4
MSR-VTT full split - TeachVideo comparison with other methods. As it can be seen, if we remove a video modality from the CE+ model, the performance drops (CE+ (w/o Audio) and CE+(w/o Action). However, if we employ TeachVideo, we can regain some of the performance, even without using that modality during inference.

Model	VT PT	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
QB-Norm [73]	✓	t2v	29.6	54.5	65.3	4.0	v2t	–	–	–	–	–
CLIP2Video [74]	✓	t2v	29.8	55.5	66.2	4.0	v2t	54.6	82.1	90.8	1.0	–
Dual [75]	✗	t2v	7.7	22.0	31.8	32.0	v2t	13.0	30.8	43.3	15.0	–
HGR [20]	✗	t2v	9.2	26.2	36.5	24.0	v2t	15.0	36.7	48.8	11.0	–
MoEE [5]	✗	t2v	$11.1_{\pm0.1}$	$30.7_{\pm0.1}$	$42.9_{\pm0.1}$	$15.0_{\pm0.0}$	v2t	$16.5_{\pm0.1}$	$43.1_{\pm0.5}$	$57.3_{\pm0.6}$	$7.7_{\pm0.5}$	400.41*M*
CE [6]	✗	t2v	$11.0_{\pm0.0}$	$30.8_{\pm0.1}$	$43.3_{\pm0.3}$	$15.0_{\pm0.0}$	v2t	$17.0_{\pm0.5}$	$43.5_{\pm0.4}$	$57.8_{\pm0.5}$	$7.2_{\pm0.2}$	183.45*M*
QB-Norm [73]	✗	t2v	$17.3_{\pm0.0}$	$42.1_{\pm0.1}$	$54.9_{\pm0.1}$	$8.0_{\pm0.0}$	v2t	–	–	–	–	262.73*M*
Teacher (CE+)	✗	t2v	$13.7_{\pm0.3}$	$36.4_{\pm0.5}$	$48.3_{\pm0.5}$	$11.0_{\pm0.0}$	v2t	$22.0_{\pm0.6}$	$50.8_{\pm1.1}$	$64.5_{\pm1.1}$	$5.3_{\pm0.6}$	262.73*M*
CE+ (w/o Audio)	✗	t2v	$12.4_{\pm0.2}$	$33.6_{\pm0.6}$	$46.3_{\pm0.6}$	$12.7_{\pm0.6}$	v2t	$19.6_{\pm0.7}$	$46.3_{\pm1.0}$	$60.2_{\pm0.9}$	$6.5_{\pm0.5}$	226.3*M*
TV-CE+ (w/o Audio)	✗	t2v	$12.8_{\pm0.1}$	$34.3_{\pm0.1}$	$47.0_{\pm0.1}$	$12.0_{\pm0.0}$	v2t	$21.1_{\pm0.5}$	$49.6_{\pm0.9}$	$62.8_{\pm1.0}$	$5.8_{\pm0.3}$	226.3*M*
CE+ (w/o Action)	✗	t2v	$13.5_{\pm0.1}$	$36.0_{\pm0.1}$	$48.8_{\pm0.1}$	$11.0_{\pm0.0}$	v2t	$21.7_{\pm0.7}$	$51.4_{\pm1.1}$	$65.1_{\pm0.6}$	$5.0_{\pm0.0}$	226.69*M*
TV-CE+ (w/o Action)	✗	t2v	$14.0_{\pm0.0}$	$36.6_{\pm0.1}$	$49.6_{\pm0.0}$	$11.0_{\pm0.0}$	v2t	$22.7_{\pm0.2}$	$51.7_{\pm0.8}$	$65.8_{\pm0.6}$	$5.0_{\pm0.0}$	226.69*M*

the effectiveness of distillation. In Fig. 12b we observe that if we remove the *audio* modality from the student, but retain it for the teacher then TeachVideo improves the student but does not enable it to match the performance of the teacher. However, we find that when repeating this procedure with *Action(KN)*, TeachVideo we can successfully train a student that does not have access to this modality but nevertheless has the same performance as if it were available. Thus, the benefits of TeachVideo depend on the removed modality. Additionally, in Table 4 we present a comparison with other methods on the MSRVTT dataset. As it can be seen, if the computational resources during inference are scarce, TeachVideo can be employed to reduce the computational cost, while it may suffer from a slight decrease in performance. This computational cost reduction happens due to two factors: 1) TeachVideo requires extracting fewer video modalities for inference (in this case 6 vs 7 in the teacher case) and 2) it has fewer parameters.

5.9. Error analysis

The purpose of this section is to better the limitations and characteristics of our retrieval system. We begin with a simple unimodal analysis that investigates the influence of query length. As seen in Fig. 13a, there is a positive correlation between query length and performance, both with and without using TeachText.

As part of our qualitative inspection of the queries removed during the denoising step (described in Sec. 5.7), we observed that many of the removed sentences are quite general in nature (e.g. *"a man walks away", "a video showing scenes from a movie"*). This motivated an investigation of differences in distribution between the pool of kept sentences and the pool of removed sentences. First, as seen in Table 3, we found that retained sentences tend to be longer. Second, we used the sentence specificity estimation method of Ko et al. [72] to estimate a specificity score for each query. We found a small difference in the specificity score between removed and retained queries. One consequence of this observation is that it may be possible to predict retrieval performance to some degree for a given query by only inspecting the query itself.

Gains due to distillation. In Fig. 13b we visualise the proportions of overlapping correctly retrieved samples in terms of R1 on the test set of the MSR-VTT dataset for the teacher, the student, and the student with TeachText. We illustrate common retrievals when distillation employs three teachers. We observe that the TeachText model retains most of the knowledge of the student, but also acquires additional information from the teacher.

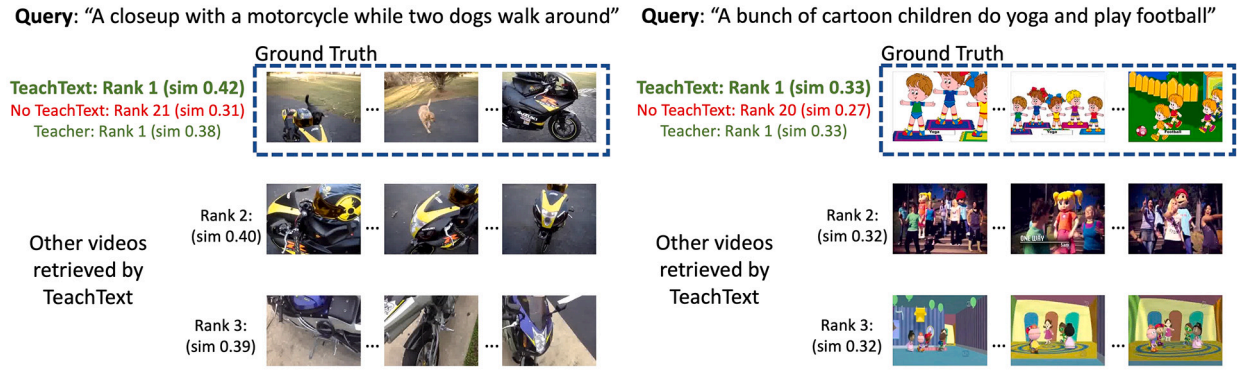


Fig. 14. Qualitative results. We present the top 3 video retrievals for each query, given by the TEACHTEXT method used on top of a CE+ architecture. Moreover, we show the rank and similarity for the teacher, as well as for the student without using TEACHTEXT for the ground truth video. We mark in green cases where the retrieval is correct in terms of R1 and with red cases where is incorrect. For each of the cases shown, the model learns from the teacher to correct its prediction.

Table 5
Condensed Movies Dataset validation split results. Results are reported for our best single model in order to enable future comparisons. *(pt) denotes pretraining on HowTo100M.

Model	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$
CMD baseline [3]	3.44	10.29	15.7	-
MMCM [76](pt)*	$5.8_{\pm 0.2}$	$15.8_{\pm 0.2}$	$22.4_{\pm 0.1}$	$73.7_{\pm 1.7}$
CE+	$\mathbf{7.2}_{\pm 0.1}$	$18.6_{\pm 0.1}$	$26.9_{\pm 0.3}$	$42.0_{\pm 1.4}$
TT-CE+	$\mathbf{7.2}_{\pm 0.4}$	$\mathbf{19.4}_{\pm 0.1}$	$\mathbf{27.8}_{\pm 0.8}$	$\mathbf{38.7}_{\pm 1.2}$

Table 6
MSR-VTT full split: Comparison to state of the art. *VT PT* marks cases where large scale visual-text pretraining was used. We show in bold the best performing model where no visual-text pretraining is used.

Model	VT PT	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
QB-Norm [73]	✓	t2v	29.6	54.5	65.3	4.0	v2t	–	–	–	–	–
CLIP2Video [74]	✓	t2v	29.8	55.5	66.2	4.0	v2t	54.6	82.1	90.8	1.0	–
Dual [75]	✗	t2v	7.7	22.0	31.8	32.0	v2t	13.0	30.8	43.3	15.0	–
HGR [20]	✗	t2v	9.2	26.2	36.5	24.0	v2t	15.0	36.7	48.8	11.0	–
MoEE [5]	✗	t2v	$11.1_{\pm 0.1}$	$30.7_{\pm 0.1}$	$42.9_{\pm 0.1}$	$15.0_{\pm 0.0}$	v2t	$16.5_{\pm 0.1}$	$43.1_{\pm 0.5}$	$57.3_{\pm 0.6}$	$7.7_{\pm 0.5}$	400.41*M*
CE [6]	✗	t2v	$11.0_{\pm 0.0}$	$30.8_{\pm 0.1}$	$43.3_{\pm 0.3}$	$15.0_{\pm 0.0}$	v2t	$17.0_{\pm 0.5}$	$43.5_{\pm 0.4}$	$57.8_{\pm 0.5}$	$7.2_{\pm 0.2}$	183.45*M*
QB-Norm [73]	✗	t2v	$17.3_{\pm 0.0}$	$42.1_{\pm 0.1}$	$54.9_{\pm 0.1}$	$\mathbf{8.0}_{\pm 0.0}$	v2t	–	–	–	–	262.73*M*
TT-CE	✗	t2v	$11.8_{\pm 0.1}$	$32.7_{\pm 0.1}$	$45.3_{\pm 0.1}$	$13.0_{\pm 0.0}$	v2t	$19.3_{\pm 0.4}$	$47.0_{\pm 0.7}$	$60.0_{\pm 0.4}$	$6.7_{\pm 0.5}$	183.45*M*
TT-CE-L	✗	t2v	$13.0_{\pm 0.0}$	$34.6_{\pm 0.1}$	$47.3_{\pm 0.2}$	$12.0_{\pm 0.0}$	v2t	$22.4_{\pm 0.3}$	$50.4_{\pm 0.6}$	$63.8_{\pm 0.3}$	$5.3_{\pm 0.5}$	66.72*M*
TT-CE+	✗	t2v	$15.0_{\pm 0.1}$	$38.5_{\pm 0.1}$	$51.7_{\pm 0.1}$	$10.0_{\pm 0.0}$	v2t	$25.3_{\pm 0.1}$	$55.6_{\pm 0.0}$	$68.6_{\pm 0.4}$	$\mathbf{4.0}_{\pm 0.0}$	262.73*M*
TT-CE-J	✗	t2v	$16.1_{\pm 0.1}$	$40.1_{\pm 0.1}$	$52.9_{\pm 0.1}$	$9.0_{\pm 0.0}$	v2t	$\mathbf{26.7}_{\pm 0.1}$	$\mathbf{57.4}_{\pm 0.1}$	$\mathbf{70.5}_{\pm 0.4}$	$\mathbf{4.0}_{\pm 0.0}$	330.28*M*
TT-CE-J(+QB-Norm [73])	✗	t2v	$\mathbf{18.5}_{\pm 0.1}$	$\mathbf{43.4}_{\pm 0.1}$	$\mathbf{55.9}_{\pm 0.0}$	$\mathbf{8.0}_{\pm 0.0}$	v2t	–	–	–	–	330.28*M*

5.10. Condensed movies challenge

TEACHTEXT formed the backbone of our submission to the 2021 Condensed Movies Challenge,[8] where we won first place. Our final submission comprised an ensemble of several models, making use of multiple text embeddings in conjunction with TEACHTEXT. In Table 5, in order to enable future comparisons, we present the results of our single models on the validation split of the Condensed Movies Dataset [3] used for the challenge.

5.11. Comparison to prior work

In Table 6, 7, 8, 9, 10, 11, 12, 13 we compare our approach to prior work on seven datasets. In order to enable fair comparisons, we split the methods into two categories: those that use large-scale pretraining on text-video data and those that do not use. We use *VT PT* column in all tables to distinguish between the two. Since in our case the focus is the usage of multiple text embeddings, we report results only on the case where the methods do not use video-text pre-training. This makes it easier to study the effect of each text embedding since there is no pre-learnt connection between the text embedding and the video embedding. Moreover, in order to

[8] https://www.robots.ox.ac.uk/~vgg/research/condensed-movies/challenge.html.

Table 7
MSR-VTT 1k-A split [18]: Comparison with others. *VT PT* marks cases where large scale visual-text pretraining was used. We show in bold the best performing model where no visual-text pretraining is used.

Model	VT PT	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
MMT [7]	✓	t2v	26.6	57.1	69.6	4.0	v2t	27.0	57.5	69.7	3.7	133.36*M*
MMCM [76]	✓	t2v	$28.7_{\pm 0.7}$	$59.5_{\pm 0.7}$	$70.3_{\pm 0.7}$	$3.8_{\pm 0.2}$	v2t	–	–	–	–	–
SSB [4]	✓	t2v	30.1	58.5	69.3	3.0	v2t	28.5	58.6	71.6	3.0	–
Frozen [77]	✓	t2v	31.0	59.5	70.5	3.0	v2t	–	–	–	–	180*M*
HD-VILA [78]	✓	t2v	35.6	65.3	78.0	3.0	v2t	–	–	–	–	–
BVTR [79]	✓	t2v	37.6	64.8	75.1	3.0	v2t	–	–	–	–	–
OA-Trans [80]	✓	t2v	40.9	70.4	80.3	2.0	v2t	–	–	–	–	–
CLIP4Clip [81]	✓	t2v	44.5	71.4	81.6	2.0	v2t	42.7	70.9	80.6	2.0	–
CLIP2Video [74]	✓	t2v	45.6	72.6	81.7	2.0	v2t	43.5	72.3	82.1	2.0	–
QB-Norm [73]	✓	t2v	47.2	73.0	83.0	2.0	v2t	–	–	–	–	–
MoEE [5]	✗	t2v	$21.6_{\pm 1.0}$	$50.8_{\pm 1.1}$	$65.6_{\pm 0.7}$	$5.3_{\pm 0.6}$	v2t	$22.4_{\pm 0.5}$	$51.2_{\pm 1.0}$	$66.1_{\pm 0.4}$	$5.2_{\pm 0.3}$	400.41*M*
CE [6]	✗	t2v	$21.7_{\pm 1.3}$	$51.8_{\pm 0.5}$	$65.7_{\pm 0.6}$	$5.0_{\pm 0.0}$	v2t	$22.7_{\pm 0.4}$	$51.8_{\pm 0.4}$	$65.7_{\pm 0.2}$	$5.0_{\pm 0.0}$	183.45*M*
MMCM [76]	✗	t2v	$22.5_{\pm 0.9}$	$53.2_{\pm 1.5}$	$67.1_{\pm 0.4}$	$4.7_{\pm 0.5}$	v2t	–	–	–	–	–
MMT [7]	✗	t2v	$24.6_{\pm 0.4}$	$54.0_{\pm 0.2}$	$67.1_{\pm 0.5}$	$4.0_{\pm 0.0}$	v2t	$24.4_{\pm 0.5}$	$56.0_{\pm 0.9}$	$67.8_{\pm 0.3}$	$4.0_{\pm 0.0}$	133.36*M*
SSB [24]	✗	t2v	27.4	56.3	67.7	**3.0**	v2t	26.6	55.1	67.5	**3.0**	–
QB-Norm [73]	✗	t2v	$\mathbf{33.3}_{\pm 0.7}$	$\mathbf{63.7}_{\pm 0.1}$	$76.3_{\pm 0.4}$	$\mathbf{3.0}_{\pm 0.0}$	v2t	–	–	–	–	262.73*M*
TT-MMT	✗	t2v	$24.8_{\pm 0.2}$	$55.9_{\pm 0.7}$	$68.5_{\pm 1.0}$	$4.3_{\pm 0.5}$	v2t	$25.1_{\pm 1.0}$	$57.1_{\pm 0.8}$	$69.9_{\pm 1.1}$	$4.0_{\pm 0.0}$	133.36*M*
TT-CE-L	✗	t2v	$26.5_{\pm 0.4}$	$58.0_{\pm 0.8}$	$71.1_{\pm 0.4}$	$4.0_{\pm 0.0}$	v2t	$27.6_{\pm 0.8}$	$58.0_{\pm 0.6}$	$70.0_{\pm 0.5}$	$4.0_{\pm 0.0}$	66.72*M*
TT-CE+	✗	t2v	$29.6_{\pm 0.3}$	$61.6_{\pm 0.5}$	$74.2_{\pm 0.3}$	$\mathbf{3.0}_{\pm 0.0}$	v2t	$\mathbf{32.1}_{\pm 0.5}$	$62.7_{\pm 0.5}$	$75.0_{\pm 0.2}$	$\mathbf{3.0}_{\pm 0.0}$	262.73*M*
TT-CE-J	✗	t2v	$29.8_{\pm 0.1}$	$63.6_{\pm 0.5}$	$\mathbf{76.7}_{\pm 0.7}$	$\mathbf{3.0}_{\pm 0.0}$	v2t	$30.6_{\pm 0.8}$	$\mathbf{65.2}_{\pm 0.8}$	$\mathbf{77.1}_{\pm 0.5}$	$\mathbf{3.0}_{\pm 0.0}$	330.28*M*

Table 8
MSVD: Comparison to state of the art methods. *VT PT* marks cases where large scale visual-text pretraining was used. We show in bold the best performing model where no visual-text pretraining is used.

Model	VT PT	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
SSB [24]	✓	t2v	28.4	60.0	72.9	4.0	v2t	34.7	59.9	70.0	3.0	–
Frozen [77]	✓	t2v	33.7	64.7	76.3	3.0	v2t	–	–	–	–	–
OA-Trans [80]	✓	t2v	39.1	68.4	80.3	2.0	v2t	–	–	–	–	–
CLIP4Clip [81]	✓	t2v	46.2	76.1	84.6	2.0	v2t	56.6	79.7	84.3	1.0	–
CLIP4Video [74]	✓	t2v	47.0	76.8	85.9	2.0	v2t	58.7	85.6	91.6	1.0	–
QB-Norm [73]	✓	t2v	47.6	77.6	86.1	2.0	v2t	–	–	–	–	–
BVTR [79]	✓	t2v	52.0	82.8	90.0	1.0	v2t	–	–	–	–	–
VSE++ [82]	✗	t2v	15.4	39.6	53.0	9.0	v2t	21.2	43.4	52.2	9.0	–
M-Cues [17]	✗	t2v	20.3	47.8	61.1	6.0	v2t	**31.5**	51.0	61.5	5.0	–
MoEE [5]	✗	t2v	$21.1_{\pm 0.2}$	$52.0_{\pm 0.7}$	$66.7_{\pm 0.2}$	$5.0_{\pm 0.0}$	v2t	$27.3_{\pm 0.9}$	$55.1_{\pm 1.2}$	$65.0_{\pm 0.8}$	$4.3_{\pm 0.5}$	131.37*M*
CE [6]	✗	t2v	$21.5_{\pm 0.5}$	$52.3_{\pm 0.8}$	$67.5_{\pm 0.7}$	$5.0_{\pm 0.0}$	v2t	$26.3_{\pm 1.4}$	$53.7_{\pm 0.4}$	$65.3_{\pm 1.1}$	$4.8_{\pm 0.2}$	84.04*M*
SSB [24]	✗	t2v	23.0	52.8	65.8	5.0	v2t	27.3	50.7	60.8	5.0	–
QB-Norm [73]	✗	t2v	$28.9_{\pm 0.3}$	$62.0_{\pm 0.4}$	$74.8_{\pm 0.3}$	$\mathbf{3.0}_{\pm 0.0}$	v2t	–	–	–	–	87.79*M*
TT-CE	✗	t2v	$22.1_{\pm 0.4}$	$52.2_{\pm 0.5}$	$67.2_{\pm 0.6}$	$5.0_{\pm 0.0}$	v2t	$26.0_{\pm 0.4}$	$53.3_{\pm 0.4}$	$63.9_{\pm 0.1}$	$4.9_{\pm 0.1}$	84.04*M*
TT-CE-L	✗	t2v	$22.5_{\pm 0.0}$	$53.7_{\pm 0.3}$	$68.7_{\pm 0.5}$	$5.0_{\pm 0.0}$	v2t	$25.6_{\pm 0.2}$	$55.7_{\pm 0.9}$	$65.9_{\pm 0.5}$	$\mathbf{4.0}_{\pm 0.0}$	27.78*M*
TT-CE+	✗	t2v	$25.4_{\pm 0.3}$	$56.9_{\pm 0.4}$	$71.3_{\pm 0.2}$	$4.0_{\pm 0.0}$	v2t	$27.1_{\pm 0.4}$	$55.3_{\pm 1.0}$	$67.1_{\pm 0.2}$	$\mathbf{4.0}_{\pm 0.0}$	87.79*M*
TT-CE-J	✗	t2v	$25.9_{\pm 0.2}$	$58.1_{\pm 0.1}$	$72.8_{\pm 0.3}$	$4.0_{\pm 0.0}$	v2t	$29.3_{\pm 0.6}$	$\mathbf{56.7}_{\pm 1.4}$	$\mathbf{68.8}_{\pm 1.2}$	$\mathbf{4.0}_{\pm 0.0}$	108.47*M*
TT-CE-J(+QB-Norm [73])	✗	t2v	$\mathbf{30.6}_{\pm 0.2}$	$\mathbf{63.6}_{\pm 0.2}$	$\mathbf{76.5}_{\pm 0.2}$	$\mathbf{3.0}_{\pm 0.0}$	v2t	–	–	–	–	108.47*M*

be as fair as possible, in each comparison we included the results of our TEACHTEXT (abbreviated TT in the tables) applied also to the best existing method for that dataset that does not use text-video data for pre-training. In this way, the architecture and the used features are identical during inference (e.g. TT-CE has the same architecture and uses the same video and text embeddings as CE). We also report results combining our strongest model with QB-Norm [73] for all datasets. Following [73] we applied the QB-Norm method on the text-to-video task. We highlight in bold the best performing method.

In Table 6, 7, 8, 9, 10, 11, 12, 13 we conduct an extensive comparison of our method with other methods from the literature for t2v and v2t tasks. We also present the number of parameters of each method where available. We observe that TEACHTEXT brings a clear improvement, while the total number of parameters remains the same as for the base architecture. We also introduce new CE-L and CE-J baselines. These are similar to CE [6] and CE+, but CE-L uses w2v as text embedding and CE-J uses *gpt-J* as text embedding. CE-L architecture has a reduced number of parameters. We observe that these architectures combine well with TEACHTEXT, showcasing the effectiveness of TEACHTEXT across different parameter regimes. Lastly, qualitative results are provided in Fig. 14.

Table 9
DiDeMo: Comparison to state of the art methods. *VT PT* marks cases where large scale visual-text pretraining was used. We show in bold the best performing model where no visual-text pretraining is used.

Model	VT PT	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$R@50\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$R@50\uparrow$	$MdR\downarrow$	Params
HD-VILA [78]	✓	t2v	28.8	57.4	68.1	–	4.0	v2t	–	–	–	–	–	–
Frozen [77]	✓	t2v	31.0	59.8	72.4	–	3.0	v2t	–	–	–	–	–	–
OA-Trans [80]	✓	t2v	34.8	64.4	75.1	–	3.0	v2t	–	–	–	–	–	–
BVTR [79]	✓	t2v	37.0	62.2	73.9	–	3.0	v2t	–	–	–	–	–	–
QB-Norm [73]	✓	t2v	43.3	71.4	80.8	–	2.0	v2t	–	–	–	–	–	–
CLIP4Clip [81]	✓	t2v	43.4	70.2	80.6	–	2.0	v2t	42.5	70.6	80.2	–	2.0	–
S2VT [83]	✗	t2v	11.9	33.6	–	76.5	13.0	v2t	13.2	33.6	–	76.5	15.0	–
FSE [84]	✗	t2v	$13.9_{\pm 0.7}$	$36.0_{\pm 0.8}$	–	$78.9_{\pm 1.6}$	$11.0_{\pm 0.0}$	v2t	$13.1_{\pm 0.5}$	$33.9_{\pm 0.4}$	–	$78.0_{\pm 0.8}$	$12.0_{\pm 0.0}$	–
MoEE [5]	✗	t2v	$16.1_{\pm 1.0}$	$41.2_{\pm 1.6}$	$55.2_{\pm 1.6}$	$81.7_{\pm 1.4}$	$8.3_{\pm 0.5}$	v2t	$16.0_{\pm 1.5}$	$41.7_{\pm 1.9}$	$54.6_{\pm 1.7}$	$81.0_{\pm 1.4}$	$8.7_{\pm 0.9}$	107.26*M*
CE [6]	✗	t2v	$17.1_{\pm 0.9}$	$41.9_{\pm 0.2}$	$56.0_{\pm 0.5}$	$83.4_{\pm 0.7}$	$8.0_{\pm 0.0}$	v2t	$17.1_{\pm 0.1}$	$41.8_{\pm 0.9}$	$55.2_{\pm 1.0}$	$83.0_{\pm 0.8}$	$7.7_{\pm 0.5}$	79.29*M*
QB-Norm [73]	✗	t2v	$24.2_{\pm 0.7}$	$50.8_{\pm 0.7}$	$64.4_{\pm 0.1}$	–	$5.3_{\pm 0.5}$	v2t	–	–	–	–	–	99.51*M*
TT-CE	✗	t2v	$21.0_{\pm 0.6}$	$47.5_{\pm 0.9}$	$61.9_{\pm 0.5}$	$86.4_{\pm 0.8}$	$6.0_{\pm 0.0}$	v2t	$20.3_{\pm 0.6}$	$46.6_{\pm 0.6}$	$59.8_{\pm 1.2}$	$85.7_{\pm 0.6}$	$6.7_{\pm 0.5}$	79.29*M*
TT-CE-L	✗	t2v	$22.3_{\pm 0.2}$	$50.1_{\pm 0.9}$	$64.3_{\pm 0.5}$	$86.9_{\pm 0.4}$	$5.3_{\pm 0.5}$	v2t	$21.3_{\pm 0.4}$	$\mathbf{48.3}_{\pm 0.5}$	$62.5_{\pm 0.3}$	$86.6_{\pm 0.1}$	$\mathbf{6.0}_{\pm 0.0}$	43.51*M*
TT-CE+	✗	t2v	$21.6_{\pm 0.7}$	$48.6_{\pm 0.4}$	$62.9_{\pm 0.6}$	$86.8_{\pm 0.3}$	$6.0_{\pm 0.0}$	v2t	$21.1_{\pm 0.2}$	$47.3_{\pm 0.2}$	$61.1_{\pm 0.4}$	$86.7_{\pm 0.2}$	$6.3_{\pm 0.5}$	99.51*M*
TT-CE-J	✗	t2v	$23.4_{\pm 0.3}$	$50.4_{\pm 0.5}$	$63.8_{\pm 0.7}$	$88.2_{\pm 0.2}$	$5.3_{\pm 0.5}$	v2t	$\mathbf{22.0}_{\pm 0.7}$	$48.0_{\pm 0.6}$	$\mathbf{62.7}_{\pm 0.8}$	$\mathbf{87.3}_{\pm 0.4}$	$\mathbf{6.0}_{\pm 0.0}$	118.81*M*
TT-CE-J(+QB-Norm [73])	✗	t2v	$\mathbf{26.2}_{\pm 0.3}$	$\mathbf{52.6}_{\pm 0.8}$	$\mathbf{66.2}_{\pm 1.1}$	$\mathbf{89.4}_{\pm 0.4}$	$\mathbf{5.0}_{\pm 0.0}$	v2t	–	–	–	–	–	118.81*M*

Table 10
LSMDC: Comparison to state of the art methods. *VT PT* marks cases where large scale visual-text pretraining was used. We show in bold the best performing model where no visual-text pretraining is used.

Model	VT PT	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
MMT [7]	✓	t2v	$12.9_{\pm 0.1}$	$29.9_{\pm 0.7}$	$40.1_{\pm 0.8}$	$19.3_{\pm 0.2}$	v2t	$12.3_{\pm 0.2}$	$28.6_{\pm 0.3}$	$38.9_{\pm 0.8}$	$20.0_{\pm 0.0}$	133.16*M*
Frozen [77]	✓	t2v	15.0	30.8	40.3	20.0	v2t	–	–	–	–	–
HD-VILA [78]	✓	t2v	17.4	34.1	44.1	15.0	v2t	–	–	–	–	–
BVTR [79]	✓	t2v	17.9	35.4	44.5	15.0	v2t	–	–	–	–	–
OA-Trans [80]	✓	t2v	18.2	34.3	43.7	18.5	v2t	–	–	–	–	–
CLIP4Clip [81]	✓	t2v	21.6	41.8	49.8	11.0	v2t	20.8	39.0	48.6	12.0	–
QB-Norm [73]	✓	t2v	22.4	40.1	49.5	11.0	v2t	–	–	–	–	–
JSFus [18]	✗	t2v	9.1	21.2	34.1	36.0	v2t	–	–	–	–	–
MoEE [5]	✗	t2v	$12.1_{\pm 0.7}$	$29.4_{\pm 0.8}$	$37.7_{\pm 0.2}$	$23.2_{\pm 0.8}$	v2t	$11.9_{\pm 0.5}$	$28.0_{\pm 0.5}$	$37.4_{\pm 0.5}$	$25.5_{\pm 1.5}$	159.78*M*
CE [6]	✗	t2v	$12.4_{\pm 0.7}$	$28.5_{\pm 0.8}$	$37.9_{\pm 0.6}$	$21.7_{\pm 0.6}$	v2t	$11.4_{\pm 0.4}$	$28.4_{\pm 0.7}$	$36.5_{\pm 0.5}$	$25.0_{\pm 0.8}$	116.86*M*
MMT [7]	✗	t2v	$13.2_{\pm 0.4}$	$29.2_{\pm 0.8}$	$38.8_{\pm 0.9}$	$21.0_{\pm 1.4}$	v2t	$12.1_{\pm 0.1}$	$29.3_{\pm 1.1}$	$37.9_{\pm 1.1}$	$22.5_{\pm 0.4}$	133.16*M*
QB-Norm [73]	✗	t2v	$17.8_{\pm 0.4}$	$37.7_{\pm 0.5}$	$47.6_{\pm 0.6}$	$12.7_{\pm 0.5}$	v2t	–	–	–	–	388.24*M*
TT-MMT	✗	t2v	$13.6_{\pm 0.5}$	$31.2_{\pm 0.4}$	$40.8_{\pm 0.5}$	$17.7_{\pm 0.5}$	v2t	$12.5_{\pm 0.3}$	$31.3_{\pm 0.6}$	$41.0_{\pm 1.1}$	$18.7_{\pm 0.5}$	133.16*M*
TT-CE-L	✗	t2v	$14.2_{\pm 0.2}$	$30.6_{\pm 0.3}$	$40.0_{\pm 0.5}$	$20.3_{\pm 0.5}$	v2t	$13.6_{\pm 0.3}$	$30.8_{\pm 0.9}$	$38.9_{\pm 0.8}$	$21.5_{\pm 0.4}$	87.22*M*
TT-CE+	✗	t2v	$17.2_{\pm 0.4}$	$36.5_{\pm 0.6}$	$46.3_{\pm 0.3}$	$13.7_{\pm 0.5}$	v2t	$17.5_{\pm 0.6}$	$36.0_{\pm 1.2}$	$45.0_{\pm 0.5}$	$14.3_{\pm 0.9}$	388.24*M*
TT-CE-J	✗	t2v	$19.6_{\pm 0.8}$	$39.6_{\pm 0.5}$	$50.1_{\pm 0.7}$	$10.7_{\pm 0.5}$	v2t	$\mathbf{19.3}_{\pm 0.4}$	$\mathbf{38.9}_{\pm 0.2}$	$\mathbf{48.4}_{\pm 0.6}$	$\mathbf{11.3}_{\pm 0.5}$	491.98*M*
TT-CE-J(+QB-Norm [73])	✗	t2v	$\mathbf{20.5}_{\pm 0.5}$	$\mathbf{40.7}_{\pm 0.3}$	$\mathbf{51.3}_{\pm 0.5}$	$\mathbf{9.7}_{\pm 0.5}$	v2t	–	–	–	–	491.98*M*

6. Conclusion

In this paper, we have introduced TeachText, a novel algorithm designed for text-video retrieval tasks. Utilizing a teacher-student paradigm, this algorithm effectively leverages multiple text embeddings, allowing for improved performance across diverse benchmarks. The key contributions of this work include the introduction of TeachText, which makes use of multiple text encoders to enhance retrieval tasks. Additionally, we present a unique approach to learning the retrieval similarity matrix, distinguishing our work from existing techniques. We also demonstrate the utility of our method for dataset denoising, contributing to improved data quality for text-video retrieval. Experimental results validate the effectiveness of our method, showing notable improvements on six different benchmarks. This research builds upon our previous publication in ICCV 2021. We have further optimized the CE+ architecture, adding two new variants that either minimize parameters or utilize GPT-J embeddings for better performance. Additional results are provided for the Condensed Movies Dataset, along with an in-depth analysis that includes ablation studies and error analysis. Our work also expands upon denoising techniques, offering a thorough evaluation of the TeachVideo approach. Overall, this paper presents significant advancements in the field of text-video retrieval, both in terms of methodological contributions and practical applications.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Table 11
ActivityNet: Comparison to state of the art methods. *VT PT* marks cases where large scale visual-text pretraining was used. We show in bold the best performing model where no visual-text pretraining is used.

Model	VT PT	Task	$R@1\uparrow$	$R@5\uparrow$	$R@50\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@50\uparrow$	$MdR\downarrow$	Params
MMT [7]	✓	t2v	$28.7_{\pm 0.2}$	$61.4_{\pm 0.2}$	$94.5_{\pm 0.0}$	$3.3_{\pm 0.5}$	v2t	$28.9_{\pm 0.2}$	$61.1_{\pm 0.2}$	$94.3_{\pm 0.4}$	$4.0_{\pm 0.0}$	$127.35M$
HD-VILA [78]	✓	t2v	28.5	57.4	94.0	4.0	v2t	–	–	–	–	–
MMCM [76]	✓	t2v	$29.0_{\pm 0.5}$	$61.7_{\pm 0.3}$	$94.6_{\pm 0.2}$	$4.0_{\pm 0.0}$	v2t	–	–	–	–	–
SSB [24]	✓	t2v	29.2	61.6	94.7	3.0	v2t	28.7	60.8	94.8	2.0	–
Frozen [77]	✓	t2v	28.8	60.9	–	3.0	v2t	–	–	–	–	–
CLIP4Clip [81]	✓	t2v	40.5	72.4	98.2	2.0	v2t	41.4	73.7	85.3	2.0	–
QB-Norm [73]	✓	t2v	41.4	71.4	97.6	2.0	v2t	–	–	–	–	–
MoEE [5]	✗	t2v	$19.7_{\pm 0.3}$	$50.0_{\pm 0.5}$	$92.0_{\pm 0.2}$	$5.3_{\pm 0.5}$	v2t	$18.3_{\pm 0.5}$	$48.3_{\pm 0.8}$	$92.0_{\pm 0.2}$	$6.0_{\pm 0.0}$	$330.42M$
CE [6]	✗	t2v	$19.9_{\pm 0.3}$	$50.1_{\pm 0.7}$	$92.2_{\pm 0.6}$	$5.3_{\pm 0.5}$	v2t	$18.6_{\pm 0.3}$	$48.6_{\pm 0.7}$	$92.0_{\pm 0.2}$	$6.0_{\pm 0.0}$	$260.68M$
HSE [57]	✗	t2v	20.5	49.3	–	–	v2t	18.7	48.1	–	–	–
MMT [7]	✗	t2v	$22.7_{\pm 0.2}$	$54.2_{\pm 1.0}$	$93.2_{\pm 0.4}$	$5.0_{\pm 0.0}$	v2t	$22.9_{\pm 0.8}$	$54.8_{\pm 0.4}$	$93.1_{\pm 0.2}$	$4.3_{\pm 0.5}$	$127.35M$
SSB [24]	✗	t2v	26.8	58.1	93.5	**3.0**	v2t	**25.5**	57.3	93.5	**3.0**	–
QB-Norm [73]	✗	t2v	$27.0_{\pm 0.2}$	$60.6_{\pm 0.4}$	$96.8_{\pm 0.0}$	$4.0_{\pm 0.0}$	v2t	–	–	–	–	$376.02M$
TT-MMT	✗	t2v	$25.0_{\pm 0.3}$	$58.7_{\pm 0.4}$	$95.6_{\pm 0.2}$	$4.0_{\pm 0.0}$	v2t	$24.4_{\pm 0.1}$	$\mathbf{58.2}_{\pm 0.3}$	$95.7_{\pm 0.1}$	$4.0_{\pm 0.0}$	$127.35M$
TT-CE-L	✗	t2v	$23.3_{\pm 0.1}$	$56.3_{\pm 0.1}$	$95.5_{\pm 0.1}$	$4.0_{\pm 0.0}$	v2t	$20.7_{\pm 0.2}$	$52.8_{\pm 0.2}$	$94.4_{\pm 0.0}$	$5.0_{\pm 0.0}$	$103M$
TT-CE+	✗	t2v	$23.5_{\pm 0.2}$	$57.2_{\pm 0.5}$	$96.1_{\pm 0.1}$	$4.0_{\pm 0.0}$	v2t	$23.0_{\pm 0.3}$	$56.1_{\pm 0.2}$	$95.8_{\pm 0.0}$	$4.0_{\pm 0.0}$	$376.02M$
TT-CE-J	✗	t2v	$24.7_{\pm 0.1}$	$58.3_{\pm 0.2}$	$96.1_{\pm 0.1}$	$4.0_{\pm 0.0}$	v2t	$23.6_{\pm 0.0}$	$56.8_{\pm 0.4}$	$\mathbf{96.1}_{\pm 0.0}$	$4.0_{\pm 0.0}$	$470.11M$
TT-CE-J(+QB-Norm [73])	✗	t2v	$\mathbf{28.3}_{\pm 0.3}$	$\mathbf{61.6}_{\pm 0.1}$	$\mathbf{96.8}_{\pm 0.1}$	$\mathbf{3.0}_{\pm 0.0}$	t2v	–	–	–	–	$470.11M$

Table 12
VaTeX: Comparison to state of the art methods. *VT PT* marks cases where large scale visual-text pretraining was used. We show in bold the best performing model where no visual-text pretraining is used.

Model	VT PT	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
SSB [24]	✓	t2v	45.9	82.4	90.4	1.0	v2t	61.2	85.2	91.8	1.0	–
CLIP2Video [74]	✓	t2v	57.3	90.0	95.5	1.0	v2t	76.0	97.7	99.9	1.0	–
QB-Norm [73]	✓	t2v	58.8	88.3	93.8	1.0	v2t	-	-	-	-	-
VSE [85]	✗	t2v	28.0	64.3	76.9	3.0	v2t	–	–	–	–	–
Dual [75]	✗	t2v	31.1	67.4	78.9	3.0	v2t	–	–	–	–	–
VSE++ [82]	✗	t2v	33.7	70.1	81.0	2.0	v2t	–	–	–	–	–
HGR [20]	✗	t2v	35.1	73.5	83.5	2.0	v2t	–	–	–	–	–
SSB [24]	✗	t2v	44.6	81.8	89.5	**1.0**	v2t	58.1	83.8	90.9	**1.0**	–
CE [6]	✗	t2v	$47.9_{\pm 0.1}$	$84.2_{\pm 0.1}$	$91.3_{\pm 0.1}$	$2.0_{\pm 0.0}$	v2t	$60.7_{\pm 1.0}$	$89.0_{\pm 0.4}$	$94.9_{\pm 0.2}$	$\mathbf{1.0}_{\pm 0.0}$	$115.56M$
QB-Norm [73]	✗	t2v	$54.8_{\pm 0.1}$	$88.2_{\pm 0.1}$	$93.8_{\pm 0.1}$	$\mathbf{1.0}_{\pm 0.0}$	v2t	–	–	–	–	$223.1M$
TT-CE	✗	t2v	$49.7_{\pm 0.1}$	$85.6_{\pm 0.1}$	$92.4_{\pm 0.1}$	$2.0_{\pm 0.0}$	v2t	$62.1_{\pm 0.2}$	$90.0_{\pm 0.1}$	$95.3_{\pm 0.1}$	$\mathbf{1.0}_{\pm 0.0}$	$115.56M$
TT-CE-L	✗	t2v	$51.5_{\pm 0.1}$	$86.5_{\pm 0.1}$	$92.6_{\pm 0.1}$	$\mathbf{1.0}_{\pm 0.0}$	v2t	$65.0_{\pm 0.5}$	$90.3_{\pm 0.5}$	$95.9_{\pm 0.2}$	$\mathbf{1.0}_{\pm 0.0}$	$55.07M$
TT-CE+	✗	t2v	$53.2_{\pm 0.2}$	$87.4_{\pm 0.1}$	$93.3_{\pm 0.0}$	$\mathbf{1.0}_{\pm 0.0}$	v2t	$64.7_{\pm 0.3}$	$91.5_{\pm 0.3}$	$96.2_{\pm 0.1}$	$\mathbf{1.0}_{\pm 0.0}$	$223.1M$
TT-CE-J	✗	t2v	$55.4_{\pm 0.2}$	$89.0_{\pm 0.1}$	$94.3_{\pm 0.0}$	$\mathbf{1.0}_{\pm 0.0}$	v2t	$\mathbf{68.7}_{\pm 0.6}$	$\mathbf{92.2}_{\pm 0.1}$	$\mathbf{96.7}_{\pm 0.1}$	$\mathbf{1.0}_{\pm 0.0}$	$281M$
TT-CE-J(+QB-Norm [73])	✗	t2v	$\mathbf{56.9}_{\pm 0.0}$	$\mathbf{89.5}_{\pm 0.1}$	$\mathbf{94.8}_{\pm 0.0}$	$\mathbf{1.0}_{\pm 0.0}$	v2t	–	–	–	–	$281M$

Table 13
QuerYD: Comparison to state of the art methods. *VT PT* marks cases where large scale visual-text pretraining was used. We show in bold the best performing model where no visual-text pretraining is used.

Model	VT PT	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Task	$R@1\uparrow$	$R@5\uparrow$	$R@10\uparrow$	$MdR\downarrow$	Params
MoEE [5]	✗	t2v	$11.6_{\pm 1.3}$	$30.2_{\pm 3.0}$	$43.2_{\pm 3.1}$	$14.2_{\pm 1.6}$	v2t	$13.0_{\pm 3.1}$	$30.9_{\pm 2.0}$	$43.0_{\pm 2.8}$	$14.5_{\pm 1.8}$	$57.75M$
CE [6]	✗	t2v	$13.9_{\pm 0.8}$	$37.6_{\pm 1.2}$	$48.3_{\pm 1.4}$	$11.3_{\pm 0.6}$	v2t	$13.7_{\pm 0.7}$	$35.2_{\pm 2.7}$	$46.9_{\pm 3.2}$	$12.3_{\pm 1.5}$	$30.82M$
QB-Norm [73]	✗	t2v	$15.1_{\pm 1.6}$	$38.3_{\pm 2.4}$	$51.2_{\pm 2.8}$	$10.3_{\pm 1.7}$	v2t	–	–	–	–	$30.82M$
TT-CE	✗	t2v	$14.2_{\pm 1.4}$	$36.6_{\pm 2.0}$	$51.1_{\pm 2.1}$	$9.7_{\pm 1.2}$	v2t	$14.1_{\pm 0.5}$	$34.8_{\pm 3.0}$	$\mathbf{49.1}_{\pm 0.3}$	$\mathbf{11.3}_{\pm 0.5}$	$30.82M$
TT-CE+	✗	t2v	$14.4_{\pm 0.5}$	$37.7_{\pm 1.7}$	$50.9_{\pm 1.6}$	$9.8_{\pm 1.0}$	v2t	$\mathbf{14.3}_{\pm 0.6}$	$\mathbf{36.3}_{\pm 0.9}$	$48.3_{\pm 1.2}$	$\mathbf{11.3}_{\pm 0.5}$	$30.82M$
TT-CE-J	✗	t2v	$13.9_{\pm 1.3}$	$38.8_{\pm 2.8}$	$51.6_{\pm 1.8}$	$10.0_{\pm 0.8}$	v2t	$13.9_{\pm 0.5}$	$36.0_{\pm 0.9}$	$48.6_{\pm 1.7}$	$\mathbf{11.3}_{\pm 1.2}$	$30.82M$
TT-CE-J(+QB-Norm [73])	✗	t2v	$\mathbf{15.4}_{\pm 0.9}$	$\mathbf{40.5}_{\pm 1.0}$	$\mathbf{52.3}_{\pm 1.7}$	$\mathbf{9.2}_{\pm 1.0}$	v2t	–	–	–	–	$30.82M$

Acknowledgements

This work was supported by EPSRC grant EP/T028572/1 and a gift from Adobe. M.L. was supported by UEFISCDI, under project EEA-RO-2018-0496. The authors would like to thank Gyungin Shin, Iulia Duta and Jenny Hu for assistance. S.A would like to acknowledge the support of Z. Novak, N. Novak and S. Carlson in enabling his contribution.

During the preparation of this work the authors used GPT-4 in order to slightly rephrase parts of the content. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Data availability

Code and data are available online as state in the paper, at https://www.robots.ox.ac.uk/vgg/research/teachtext/.

References

[1] I. Croitoru, S.-V. Bogolin, M. Leordeanu, H. Jin, A. Zisserman, S. Albanie, Y. Liu, Teachtext: crossmodal generalized distillation for text-video retrieval, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021.
[2] B. Wang, A. Komatsuzaki, GPT-J-6B: a 6 billion parameter autoregressive language model, in, https://github.com/kingoflolz/mesh-transformer-jax, 2021.
[3] M. Bain, A. Nagrani, A. Brown, A. Zisserman, Condensed movies: story based retrieval with contextual embeddings, in: Proceedings of the Asian Conference on Computer Vision, 2020.
[4] W. Park, D. Kim, Y. Lu, M. Cho, Relational knowledge distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[5] A. Miech, I. Laptev, J. Sivic, Learning a text-video embedding from incomplete and heterogeneous data, in: arXiv preprint, arXiv:1804.02516, 2018.
[6] Y. Liu, S. Albanie, A. Nagrani, A. Zisserman, Use what you have: video retrieval using representations from collaborative experts, in: arXiv preprint, arXiv:1907.13487, 2019.
[7] V. Gabeur, C. Sun, K. Alahari, C. Schmid, Multi-modal transformer for video retrieval, in: Proceedings of the European Conference on Computer Vision, 2020.
[8] R. Xu, C. Xiong, W. Chen, J.J. Corso, Jointly modeling deep video and compositional text to bridge vision and language in a unified framework, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015.
[9] J. Sivic, A. Zisserman, Video Google: a text retrieval approach to object matching in videos, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, 2003.
[10] I. Laptev, P. Pérez, Retrieving actions in movies, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2007.
[11] Y.-G. Jiang, C.-W. Ngo, J. Yang, Towards optimal bag-of-features for object categorization and semantic video retrieval, in: Proceedings of the ACM International Conference on Image and Video Retrieval, 2007.
[12] L. Jiang, D. Meng, T. Mitamura, A.G. Hauptmann, Easy samples first: self-paced reranking for zero-example multimedia search, in: Proceedings of the 22nd ACM International Conference on Multimedia, 2014.
[13] C. Gan, Y. Yang, L. Zhu, D. Zhao, Y. Zhuang, Recognizing an Action Using Its Name: A Knowledge-Based Approach, vol. 120, Springer, 2016, pp. 61–77.
[14] C. Gan, C. Sun, R. Nevatia, Deck: discovering event composition knowledge from web images for zero-shot event detection and recounting in videos, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017.
[15] Y. Aytar, M. Shah, J. Luo, Utilizing semantic word similarity measures for video retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2008.
[16] J. Dong, X. Li, C.G. Snoek, Word2visualvec: image and video to sentence matching by visual feature prediction, in: arXiv preprint, arXiv:1604.06838, 2016.
[17] N.C. Mithun, J. Li, F. Metze, A.K. Roy-Chowdhury, Learning joint embedding with multimodal cues for cross-modal video-text retrieval, in: Proceedings of the ACM International Conference on Multimedia Retrieval, 2018.
[18] Y. Yu, J. Kim, G. Kim, A joint sequence fusion model for video question answering and retrieval, in: Proceedings of the European Conference on Computer Vision, 2018.
[19] M. Wray, D. Larlus, G. Csurka, D. Damen, Fine-grained action retrieval through multiple parts-of-speech embeddings, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[20] S. Chen, Y. Zhao, Q. Jin, Q. Wu, Fine-grained video-text retrieval with hierarchical graph reasoning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020.
[21] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, J. Sivic, Howto100m: learning a text-video embedding by watching hundred million narrated video clips, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[22] A. Miech, J.-B. Alayrac, L. Smaira, I. Laptev, J. Sivic, A. Zisserman, End-to-end learning of visual representations from uncurated instructional videos, in: arXiv preprint, arXiv:1912.06430, 2019.
[23] B. Korbar, F. Petroni, R. Girdhar, L. Torresani, Video understanding as machine translation, in: arXiv preprint, arXiv:2006.07203, 2020.
[24] M. Patrick, P.-Y. Huang, Y. Asano, F. Metze, A. Hauptmann, J. Henriques, A. Vedaldi, Support-set bottlenecks for video-text representation learning, in: arXiv preprint, arXiv:2010.02824, 2020.
[25] C.G. Snoek, M. Worring, Multimodal Video Indexing: A Review of the State-of-the-Art, vol. 25, Springer, 2005, pp. 5–35.
[26] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: arXiv preprint, arXiv:1301.3781, 2013.
[27] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, et al., Improving language understanding by generative pre-training, OpenAI, 2018.
[28] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, Language models are unsupervised multitask learners, in: Preprint, 2019.
[29] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the Association for Computational Linguistics, 2019.
[30] A. Burns, R. Tan, K. Saenko, S. Sclaroff, B.A. Plummer, Language features matter: effective language representations for vision-language tasks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[31] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: a robustly optimized bert pretraining approach, in: arXiv preprint, arXiv:1907.11692, 2019.
[32] Z. Lan, M. Chen, S. Goodman, K. Gimpel, P. Sharma, R. Soricut, Albert: a lite bert for self-supervised learning of language representations, in: arXiv preprint, arXiv:1909.11942, 2019.
[33] L. Breiman, N. Shang, Born Again Trees, vol. 1, Citeseer, 1996, p. 2.
[34] C. Buciluǎ, R. Caruana, A. Niculescu-Mizil, Model compression, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006.
[35] G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network, in: arXiv preprint, arXiv:1503.02531, 2015.
[36] D. Lopez-Paz, L. Bottou, B. Schölkopf, V. Vapnik, Unifying distillation and privileged information, in: Proceedings of the International Conference on Learning Representations, 2016.
[37] V. Vapnik, A. Vashist, A New Learning Paradigm: Learning Using Privileged Information, vol. 22, Elsevier, 2009, pp. 544–557.
[38] V. Vapnik, R. Izmailov, Learning using privileged information: similarity control and knowledge transfer 16 (2015) 2023–2049.
[39] A. Romero, N. Ballas, S.E. Kahou, A. Chassang, C. Gatta, Y. Bengio, Fitnets: hints for thin deep nets, in: arXiv preprint, arXiv:1412.6550, 2014.
[40] S. Zagoruyko, N. Komodakis, Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer, in: arXiv preprint, arXiv:1612.03928, 2016.
[41] Z. Huang, N. Wang, Like what you like: knowledge distill via neuron selectivity transfer, in: arXiv preprint, arXiv:1707.01219, 2017.
[42] Y.-L. Li, X. Liu, X. Wu, Y. Li, C. Lu, Hoi analysis: integrating and decomposing human-object interaction, in: H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, H. Lin (Eds.), Advances in Neural Information Processing Systems, vol. 33, Curran Associates, Inc., 2020, pp. 5011–5022.

[43] Y. Cao, M. Long, J. Wang, S. Liu, Deep visual-semantic quantization for efficient image retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.
[44] Q. Wang, A.B. Chan, Describing like humans: on diversity in image captioning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[45] P. Bojanowski, R. Lajugie, E. Grave, F. Bach, I. Laptev, J. Ponce, C. Schmid, Weakly-supervised alignment of video with text, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2015.
[46] R. Socher, A. Karpathy, Q.V. Le, C.D. Manning, A.Y. Ng, Grounded Compositional Semantics for Finding and Describing Images with Sentences, vol. 2, MIT Press, 2014, pp. 207–218.
[47] J. Carreira, A. Zisserman, Quo vadis, action recognition? A new model and the kinetics dataset, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.
[48] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.
[49] D. Ghadiyaram, D. Tran, D. Mahajan, Large-scale weakly-supervised pre-training for video action recognition, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[50] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: an imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019.
[51] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: arXiv preprint, arXiv:1412.6980, 2014.
[52] J. Xu, T. Mei, T. Yao, Y. Rui, Msr-vtt: a large video description dataset for bridging video and language, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016.
[53] D.L. Chen, W.B. Dolan, Collecting highly parallel data for paraphrase evaluation, in: Proceedings of the Association for Computational Linguistics, 2011.
[54] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell, K. Saenko, Sequence to sequence-video to text, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2015.
[55] L. Anne Hendricks, O. Wang, E. Shechtman, J. Sivic, T. Darrell, B. Russell, Localizing moments in video with natural language, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.
[56] B. Thomee, D.A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, L.-J. Li, Yfcc100m: The New Data in Multimedia Research, vol. 59, ACM, New York, NY, USA, 2016, pp. 64–73.
[57] B. Zhang, H. Hu, F. Sha, Cross-modal and hierarchical modeling of video and text, in: Proceedings of the European Conference on Computer Vision, 2018.
[58] A. Rohrbach, A. Torabi, M. Rohrbach, N. Tandon, C. Pal, H. Larochelle, A. Courville, B. Schiele, Movie Description, vol. 123, Springer, 2017, pp. 94–120.
[59] F. Caba Heilbron, V. Escorcia, B. Ghanem, J. Carlos Niebles, Activitynet: a large-scale video benchmark for human activity understanding, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2015.
[60] X. Wang, J. Wu, J. Chen, L. Li, Y.-F. Wang, W.Y. Wang, Vatex: a large-scale, high-quality multilingual dataset for video-and-language research, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[61] A.-M. Oncescu, J.F. Henriques, Y. Liu, A.Z. Zisserman, S. Albanie, Queryd: a video dataset with high-quality textual and audio narrations, in: arXiv preprint, arXiv:2011.11071, 2020.
[62] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, 2019, Squeeze-and-excitation networks.
[63] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.
[64] D. Mahajan, R. Girshick, V. Ramanathan, K. He, M. Paluri, Y. Li, A. Bharambe, L. van der Maaten, Exploring the limits of weakly supervised pretraining, in: Proceedings of the European Conference on Computer Vision, 2018.
[65] S. Hershey, S. Chaudhuri, D.P.W. Ellis, J.F. Gemmeke, A. Jansen, C. Moore, M. Plakal, D. Platt, R.A. Saurous, B. Seybold, M. Slaney, R. Weiss, K. Wilson, Cnn architectures for large-scale audio classification, in: International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2017.
[66] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: A 10 million image database for scene recognition, 2017.
[67] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2017.
[68] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D.A. Shamma, et al., Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations, vol. 123, Springer, 2017, pp. 32–73.
[69] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, in: Advances in Neural Information Processing Systems, 2017.
[70] F. team, Ffmpeg, https://www.ffmpeg.org/, 2020.
[71] J. Tang, K. Wang, Ranking distillation: learning compact ranking models with high performance for recommender system, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018.
[72] W.-J. Ko, G. Durrett, J.J. Li, Domain agnostic real-valued specificity prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019.
[73] S.-V. Bogolin, I. Croitoru, H. Jin, Y. Liu, S. Albanie, Cross modal retrieval with querybank normalisation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[74] H. Fang, P. Xiong, L. Xu, Y. Chen, Clip2video: mastering video-text retrieval via image clip, in: arXiv preprint, arXiv:2106.11097, 2021.
[75] J. Dong, X. Li, C. Xu, S. Ji, X. Wang, Dual dense encoding for zero-example video retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019.
[76] V. Gabeur, A. Nagrani, C. Sun, K. Alahari, C. Schmid, Masking modalities for cross-modal video retrieval, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2022.
[77] M. Bain, A. Nagrani, G. Varol, A. Zisserman, Frozen in time: a joint video and image encoder for end-to-end retrieval, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021.
[78] H. Xue, T. Hang, Y. Zeng, Y. Sun, B. Liu, H. Yang, J. Fu, B. Guo, Advancing high-resolution video-language representation with large-scale video transcriptions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[79] Y. Ge, Y. Ge, X. Liu, D. Li, Y. Shan, X. Qie, P. Luo, Bridging video-text retrieval with multiple choice questions, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[80] J. Wang, Y. Ge, G. Cai, R. Yan, X. Lin, Y. Shan, X. Qie, M.Z. Shou, Object-aware video-language pre-training for retrieval, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022.
[81] H. Luo, L. Ji, M. Zhong, Y. Chen, W. Lei, N. Duan, T. Li, CLIP4Clip: an empirical study of clip for end to end video clip retrieval, in: arXiv preprint, arXiv: 2104.08860, 2021.
[82] F. Faghri, D.J. Fleet, J.R. Kiros, S. Fidler, Vse++: improving visual-semantic embeddings with hard negatives, in: arXiv preprint, arXiv:1707.05612, 2017.
[83] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, K. Saenko, Translating videos to natural language using deep recurrent neural networks, in: arXiv preprint, arXiv:1412.4729, 2014.
[84] M. Zhang, J. Lucas, J. Ba, G.E. Hinton, Lookahead optimizer: k steps forward, 1 step back, in: Advances in Neural Information Processing Systems, 2019.
[85] R. Kiros, R. Salakhutdinov, R.S. Zemel, Unifying visual-semantic embeddings with multimodal neural language models, in: arXiv preprint, arXiv:1411.2539, 2014.