Condensed Movies: Story Based Retrieval with Contextual Embeddings

Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman

Visual Geometry Group, Department of Engineering Science, University of Oxford
{maxbain,arsha,abrown,az}@robots.ox.ac.uk

Abstract. Our objective in this work is long range understanding of the narrative structure of movies. Instead of considering the entire movie, we propose to learn from the 'key scenes' of the movie, providing a `condensed` look at the full storyline. To this end, we make the following three contributions: (i) We create the `Condensed Movies Dataset (CMD)` consisting of the key scenes from over 3K movies: each key scene is accompanied by a high level semantic description of the scene, character face-tracks, and metadata about the movie. The dataset is scalable, obtained automatically from YouTube, and is freely available for anybody to download and use. It is also an order of magnitude larger than existing movie datasets in the number of movies; (ii) We provide a deep network baseline for text-to-video retrieval on our dataset, combining character, speech and visual cues into a single video embedding; and finally (iii) We demonstrate how the addition of context from other video clips improves retrieval performance.

1 Introduction

Imagine you are watching the movie *'Trading Places'*, and you want to instantly fast forward to a scene, one where 'Billy reveals the truth to Louis about the Duke's bet, a bet which changed both their lives'. In order to solve this task automatically, an intelligent system would need to watch the movie up to this point, have knowledge of Billy, Louis and the Duke's identities, understand that the Duke made a bet, and know the outcome of this bet (Fig. 1). This high level understanding of the movie narrative requires knowledge of the characters' identities, their relationships, motivations and conversations, and ultimately their behaviour. Since movies and TV shows can provide an ideal source of data to test this level of story understanding, there have been a number of movie related datasets and tasks proposed by the computer vision community [1,2,3,4,5].

However, despite the recent proliferation of movie-related datasets, high level semantic understanding of human narratives still remains a challenging task. There are a number of reasons for this lack of progress: (i) semantic annotation is expensive and challenging to obtain, inherently restricting the size of current movie datasets to only hundreds of movies, and often, only part of the movie is annotated in detail [1,2,3]; (ii) movies are very long (roughly 2 hours) and video architectures struggle to learn over such large timescales; (iii) there are legal and copyright issues surrounding a majority of these datasets [1,3], which hinder their widespread availability and adoption in the community; and finally (iv) the subjective nature of the task makes it difficult to define objectives and metrics [**?**].

A number of different works have recently creatively identified that certain domains of videos, such as narrated instructional videos [6,7,8] and lifestyle vlogs [9,10] are available

Fig. 1: **Condensed Movies:** The dataset consists of the key scenes in a movie (ordered by time), together with high level semantic descriptions. Note how the caption of a scene (far right) is based on the knowledge of past scenes in the movie – one where the Dukes exchange money to settle their bet (highlighted in yellow), and another scene showing their lives before the bet, homeless and pan-handling (highlighted in green).

in large numbers on YouTube and are a good source of supervision for video-text models as the speech describes the video content. In a similar spirit, videos from the MovieClips channel on YouTube[1], which contains the key scenes or clips from numerous movies, are also accompanied by a semantic text description describing the content of each clip.

Our first objective in this paper is to curate a dataset, suitable for learning and evaluating long range narrative structure understanding, from the available video clips and associated annotations of the MovieClips channel. To this end, we curate a dataset of 'condensed' movies, called the Condensed Movie Dataset (CMD) which provides a *condensed* snapshot into the entire storyline of a movie. In addition to just the video, we also download and clean the high level semantic descriptions accompanying each key scene that describes characters, their motivations, actions, scenes, objects, interactions and relationships. We also provide labelled face-tracks of the principal actors (generated automatically), as well as the metadata associated with the movie (such as cast lists, synopsis, year, genre). Essentially, all the information required to (sparsely) generate a MovieGraph [4]. The dataset consists of over 3000 movies.

Previous work on video retrieval and video understanding has largely treated video clips as independent entities, divorced from their context [2,11,12]. But this is not how movies are understood: the meaning and significance of a scene depends on its relationship to previous scenes. This is true also of TV series, where one episode depends on those leading up to it (the season arc); and even an online tutorial/lesson can refer to previous tutorials. These contextual videos are beneficial and sometimes even necessary for complete video understanding.

Our second objective is to explore the role of context in enabling video retrieval. We define a text-to-video retrieval task on the CMD, and extend the popular Mixture of Embedding Experts model [13], that can learn from the subtitles, faces, objects, actions and scenes, by adding a *Contextual Boost Module* that introduces information from past and future clips. Unlike other movie related tasks – e.g. text-to-video retrieval on the LSMDC dataset [2] or graph retrieval on the MovieQA [14] dataset that ignore identities, we also introduce a character embedding module which allows the model to reason about the identities of characters present in each clip and description. Applications

[1] https://www.youtube.com/user/movieclips

of this kind of story-based retrieval include semantic search and indexing of movies as well as intelligent fast forwards. The CMD dataset can also be used for semantic video summarization and automatic description of videos for the visually impaired (Descriptive Video Services (DVS) are currently available at a huge manual cost).

Finally, we also show preliminary results for aligning the semantic captions to the plot summaries of each movie, which places each video clip in the larger context of the movie as a whole. Data, code, models and features can be found at `https://www.robots.ox.ac.uk/~vgg/research/condensed-movies/`.

2 Related Work

Video Understanding from Movies: There is an increasing effort to develop video understanding techniques that go beyond action classification from cropped, short temporal snippets [15,16,17], to learning from longer, more complicated videos that promise a higher level of abstraction [18,19,6,20]. Movies and TV shows provide an ideal test bed for learning long-term stories, leading to a number of recent datasets focusing exclusively on this domain [1,14,2,3]. Early works, however, focused on using film and TV to learn human identity [21,22,23,24,25,26] or human actions [27,28,29,30,31] from the scripts or captions accompanying movies. Valuable recent works have proposed story-based tasks such as the visualization and grouping of scenes which belong to the same story threads [32,33], the visualization of TV episodes as a chart of character interactions [1], and more recently, the creation of more complicated movie graphs (MovieGraphs [4] is the most exhaustively annotated movie dataset to date). Such graphs have enabled explicit learning of interactions and relationships [34] between characters. This requires understanding multiple factors such as human communication, emotions, motivation, scenes and other factors that affect behavior. There has also been a recent interest in evaluating story understanding through visual question answering [14] and movie scene segmentation [33]. In contrast, we propose to evaluate story understanding through the task of text-to-video retrieval, from a set of key scenes in a movie that condense most of the salient parts of the storyline. Unlike retrieval through a complex graph [4], retrieval via text queries can be a more intuitive way for a human to interact with an intelligent system, and might help avoid some of the biases present inherently in VQA datasets [35].

Comparison to other Movie Datasets: Existing movie datasets often consist of short clips spanning entire, full length movies (which are subject to copyright and difficult for public release to the community). All such datasets also depend on exhaustive annotation, which limit their scale to hundreds of movies. Our dataset, in contrast, consists of only the key scenes from movies matched with high quality, high level semantic descriptions, allowing for a condensed look at the entire storyline. A comparison of our dataset to other datasets can be seen in Table 1.

Text-to-Video Retrieval: A common approach for learning visual embeddings from natural language supervision is to learn a joint embedding space where visual and textual cues are adjacent if they are semantically similar [13,38]. Most of these works rely on manually annotated datasets in which descriptive captions are collected for short, isolated video clips, with descriptions usually focusing on low-level visual content provided by annotators [2,12,11]. For example LSMDC [2], which is created from DVS, contains mostly low-level descriptions of the visual content in the scene, e.g. 'Abby gets in the basket', unlike the descriptions in our dataset. Most similar to our work is [39], which

Table 1: Comparison to other movie and TV show datasets. For completeness, we also compare to datasets that *only* have character ID or action annotation. 'Free' is defined here as accessible online at no cost at the time of writing. *Refers to number of TV shows.

	#Movies	#Hours	Free	Annotation Type
Sherlock [36]	1*	4		Character IDs
TVQA[37]	6*	460		VQA
AVA [16]	430	107.5	✓	Actions only
MovieGraphs [4]	51	93.9		Descriptions, graphs
MovieQA (video)[14]	140	381		VQA
MovieScenes [33]	150	250		Scene segmentations
LSMDC [2]	202	158		Captions
MSA [3]	327	516		Plots
MovieNet[5]	1,100	2,000		Plots, action tags, character IDs
CMD (Ours)	**3,605**	**1,270**	✓	Descriptions, metadata, character IDs, plots

obtains story level descriptions for shots in full movies, by aligning plot sentences to shots, and then attempting video retrieval. This, however, is challenging because often there is no shot that matches a plot sentence perfectly, and shots cover very small timescales. Unlike this work our semantic descriptions are more true to the clips themselves.

Temporal Context: The idea of exploiting surrounding context has been explored by [40], for the task of video captioning, and by [41] for video understanding. Krishna *et al.* [40] introduces a new captioning module that uses contextual information from past and future events to jointly describe all events, however this work focuses on short term context (few seconds before and after a particular clip). Wu *et al.* [41] go further, and introduce a feature bank architecture that can use contextual information over several minutes, demonstrating the performance improvements that results. Our dataset provides the opportunity to extend such feature banks (sparsely) over an entire movie.

3 Condensed Movie Dataset

We construct a dataset to facilitate machine understanding of narratives in long movies. Our dataset has the following key properties:

(1) Condensed Storylines: The video data consists of over 33,000 clips from 3,600 movies (see Table 2). For each movie there is a set of ordered clips (typically 10 or so) covering the salient parts of the film (examples can be seen in Fig. 2, top row). Each around two minutes in length, the clips contain the same rich and complex story as full-length films but an order of magnitude shorter. The distribution of video lengths in our dataset can be seen in Fig. 2 – with just the key scenes, each movie has been condensed into roughly 20 minutes each. Each clip is also accompanied by a high level description focusing on intent, emotion, relationships between characters and high level semantics (Figures 2 and 3). Compared to other video-text datasets, our descriptions are longer, and have a higher lexical diversity [42] (Table 2). We also provide face-tracks and identity labels for the main characters in each clip (Figure 2, bottom row).

(2) Online Longevity and Scalability: All the videos are obtained from the

Fig. 2: **The Condensed Movie Dataset (CMD)**. *Top:* Samples of clips and their corresponding captions from *The Karate Kid (1984)* film. In movies, as in real life, situations follow from other situations and the combination of video and text tell a concise story. Note: Every time a character is mentioned in the description, the name of the actor is present in brackets. We remove these from the figure in the interest of space. *Middle, from left to right:* Histogram of movie genres, movie release years, description length and duration of video clips. Best viewed online and zoomed in. *Bottom:* Example face-tracks labelled with the actor's name in the clips. These labels are obtained from cast lists and assigned to facetracks using our automatic labelling pipeline.

licensed, freely available YouTube channel: MovieClips[2]. We note that a common problem plaguing YouTube datasets today [43,16,15,44] is the fast shrinkage of datasets as user uploaded videos are taken down by users (over 15% of Kinetics-400 [15] is no longer available on YouTube at the time of writing, including videos from the eval sets). We believe our dataset has longevity due to the fact that the movie clips on the licensed channel are rarely taken down from YouTube. Also, this is an actively growing YouTube channel as new movies are released and added. Hence there is a potential to continually increase the size of the dataset. We note that from the period of 1st Jan 2020, to 1st September 2020, only 0.3% of videos have been removed from the YouTube channel, while an additional 2,000 videos have been uploaded, resulting in a dataset growth of 5.8% over the course of 9 months.

3.1 Dataset Collection Pipeline

In this section we describe the dataset collection pipeline.
Videos and Descriptions: Raw videos are downloaded from YouTube. Each video is

[2] `https://www.youtube.com/user/movieclips/`. Screenshots of the channel are provided in supplementary material of the ArXiV version

Table 2: Comparison to other video text retrieval datasets. MTLD is the Measure of Textual Lexical Diversity [42] for all of the descriptions in the dataset.

Dataset	#Videos/#Clips	Median caption len. (words)	MTLD	Median clip len. (secs)
MSVRTT [11]	7,180/10,000	7	26.9	15
DiDemo [12]	10,464/26,892	7	39.9	28
LSMDC [2]	200/118,114	8	61.6	5
CMD (Ours)	3,605/33,976	**18**	**89.1**	**132**

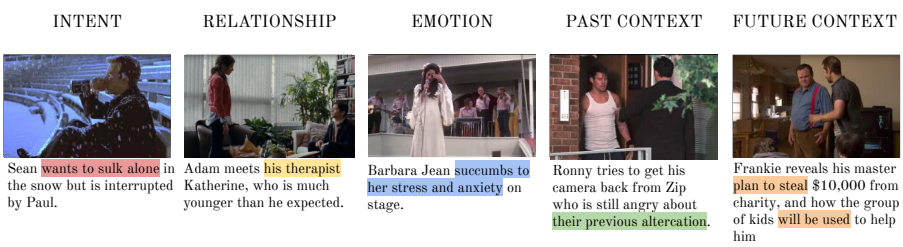

Fig. 3: **Semantic descriptions:** Examples of high level semantic descriptions accompanying each video clip in our dataset (note: actor names are removed to preserve space). Our semantic descriptions cover a number of high level concepts, including intent/motivation, relationships, emotions and attributes, and context from surrounding clips in the storyline.

accompanied by an outro at the end of the clip which contains some advertising and links to other movies. This is automatically removed by using the observation that each outro has a consistent length of either 10s (if the clip is uploaded before May 2017) or 30s if uploaded after. Approximately 1,000 videos from the channel were manually excluded from the dataset because they contained low quality descriptions or did not contain scenes from a movie. For each video, we also download the YouTube closed captions, these are a mix of high quality, human generated subtitles and automatic captions. Closed captions are missing for 36.7% of the videos. The MovieClips channel also provides a rich and high level description with each video, which we extract, clean (removing the movie title, links and advertising) and verify manually. We note that the videos also contain a watermark, usually at the bottom left of the frame. These can be easily cropped from the videos.

Metadata: For each clip, we identify its source movie by parsing the movie title from the video description and, if available, the release year (since many movies have non-unique titles). The title and release year are queried in the IMDb search engine to obtain the movie's IMDb ID, cast list and genre. IMDb identification enables correspondence to other popular movie datasets [14,45]. Plot synposes were gathered by querying the movie title and release year in the Wikipedia search engine and extracting text within the 'Plot' section of the top ranked entry. For each movie we include: (i) the movie description (short, 3-5 sentences), accompaying the video clips on the MovieClips YouTube channel; (ii) Wikipedia plot summaries (medium, 30 sentences); and (iii) IMDB plot synopses (long, 50+ sentences).

Face-tracks and Character IDs: We note that often character identities are the focal point of any storyline, and many of the descriptions reference key characters. In a similar manner to [36], we use face images downloaded from search engines to label detected and tracked faces in our dataset. Our technique involves the creation of a character embedding bank (CEB) which contains a list of characters (obtained from cast lists), and a corresponding embedding vector obtained by passing search engine image results through a deep CNN model pretrained on human faces [46]. Character IDs are then assigned to face-tracks in the video dataset when the similarity between the embeddings from the face tracks and the embeddings in the CEB (using cosine similarity) is above a certain threshold. This pipeline is described in detail in the supplementary material of the ArXiV version. We note that this is an automatic method and so does not yield perfect results, but a random manual inspection shows that it is accurate 96% of the time. Ultimately, we are able to recognize 8,375 different characters in 25,760 of the video clips.

3.2 Story Coverage

To quantitatively measure the amount of the story covered by movie clips in our dataset, we randomly sample 100 movies and manually aligned the movie clips (using the descriptions as well as the videos) to Wikipedia plot summaries (the median length of which is 32 sentences). We found that while the clips totalled only **15%** of the full-length movie in time duration, they cover **44%** of the full plot sentences, suggesting that the clips can indeed be described as key scenes. In addition, we find that the movie clips span a median range of **85.2%** of the plot, with the mean midpoint of the span being **53%**. We further show the distribution of clip sampling in Fig. **??** in the supplementary material of the ArXiV version, and find that in general there is an almost uniform coverage of the movie. While we focus on a baseline task of video-text retrieval, we also believe that the longitudindal nature of our dataset will encourage other tasks in long range movie understanding.

4 Text-to-Video Retrieval

In this section we provide a baseline task for our dataset – the task of text-to-video retrieval. The goal here is to retrieve the correct 'key scene' over all movies in the dataset, given just the high level description. Henceforth, we use the term 'video clip' to refer to one key scene, and 'description' to refer to the high level semantic text accompanying each video clip. In order to achieve this task, we learn a common embedding space for each video and the description accompanying it. More formally, if V is the video and T is the description, we learn embedding functions f and g such that the similarity $s = \langle f(V), g(T) \rangle$ is high only if T is the correct semantic description for the video V. Inspired by previous works that achieve state-of-the-art results on video retrieval tasks [13,38], we encode each video as a combination of different streams of descriptors. Each descriptor is a semantic representation of the video learnt by individual experts (that encode concepts such as scenes, faces, actions, objects and the content of conversational speech from subtitles).

Inspired by [13], we base our network architecture on a mixture of 'expert' embeddings model, wherein a separate model is learnt for each expert, which are then combined in an end-to-end trainable fashion using weights that depend on the input caption. This allows the model to learn to increase the relative weight of motion descriptors for input captions concerning human actions, or increase the relative weight of face descriptors for input captions that require detailed face understanding. We also note, however, that often the text query not only provides clues as to which expert is more valuable, but also whether it is useful to pay attention to a previous clip in the movie, by referring to something that happened previously, eg. 'Zip is *still* angry about their *previous altercation*'. Hence we introduce a Contextual Boost module (CBM), which allows the model to learn to increase the relative weight of a past video feature as well. A visual overview of the retrieval system with the CBM can be seen in Fig. 4. In regular movie datasets, the space of possible previous clips can be prohibitively large [39], however this becomes feasible with our *Condensed Movies* dataset.

Besides doing just *cross-movie* retrieval, we also adapt our model to perform *within-movie* retrieval. We note that characters are integral to a storyline, and hence for the case of within-movie retrieval, we introduce a character module, which computes a weighted one-hot vector for the characters present in the description query and another for each video clip in the dataset. We note that for cross-movie retrieval, the retrieval task becomes trivial given the knowledge of the characters in each movie, and hence to make the task more challenging (and force the network to focus on other aspects of the story), we remove the character module for this case.

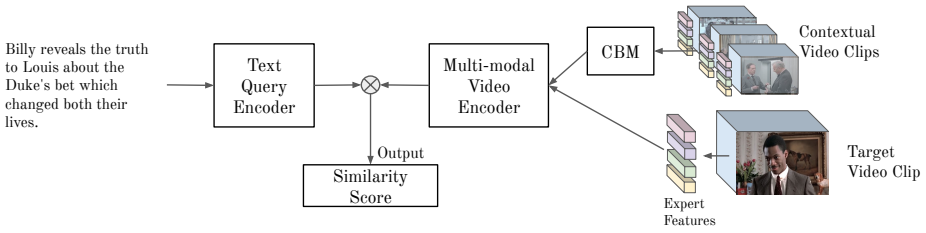

Fig. 4: **Model architecture:** An overview of text-to-video retrieval with our Contextual Boost module (CBM) that computes a similarity score between a query sentence T and a target video. CBM receives contextual video features (which are previous clips from the same movie) to improve the multimodal encoding of the target video clip. The expert features are extracted using pre-trained models for speech, motion, faces, scenes and objects.

4.1 Model Architecture

Expert Features. Stories in movies are communicated through many modalities including (but not limited to) speech, body language, facial expressions and actions. Hence we represent each input video V with K different expert streams (in our case, $K=5$ – face, subtitles, objects, motion and scene, but our framework can be extended to more experts as required).

Each input stream is denoted as I_i, where $i=1,...,K$. Adopting the approach proposed by [13], we first aggregate the descriptors of each input stream over time, using a temporal aggregation module (see Sec. 5 for details), and the resulting time-aggregated descriptor is embedded using a gated embedding module (for the precise details of the

gated embedding module, please see [13]). We then finally project each embedding to a common dimension D using a fully connected layer, giving us one expert embedding E_{V_i} for each input stream i.Hence the final output is of dimensions $K \times D$.

Text Query Encoder. The query description input is a sequence of BERT word embeddings [47] for each input sentence. These individual word embedding vectors are then aggregated into a single vector $h(T)$ representing the entire sentence using a NetVLAD [48] aggregation module. This vector $h(T)$, is used to predict the mixture weights (described in the next section). We project $h(T)$ to the same dimensions as the video expert features using the same gated embedding module followed by a fully connected layer as for the video experts (described above), once for each input source i, giving us expert embeddings E_{T_i}. Hence the final output is also of dimensions $K \times D$.

Contextual Boost Module. In both [13] and [38], the resulting expert embeddings E_{V_i} are then weighted using normalized weights $w_i(T)$ estimated from the text description T. The final similarity score s is obtained by a weighted combination of the similarity scores $s_i(E_{T_i},E_{V_i})$ between the embeddings E_{T_i} of the query sentence T and the expert embeddings E_{V_i} (obtained from the input video descriptors I_i). More formally, this is calculated as:

$$s(T,V) = \sum_{i=1}^{K} w_i(T) s_i(E_{T_i},E_{V_i}), \qquad \text{where} \qquad w_i(T) = \frac{e^{h(T)^\intercal a_i}}{\sum_{j=1}^{K} e^{h(T)^\intercal a_j}} \tag{1}$$

where s_i is the scalar product, $h(T)$ is the aggregated text query representation described above and a_i, $i=1,..,.K$ are learnt parameters used to obtained the mixture weights.

In this work, however, we extend this formulation in order to incorporate past context into the retrieval model. We would like the model to be able to predict weights for combining experts from previous clips – note we treat each expert separately in this formulation. For example, the model might want to heavily weight the subtitles from a past clip, but downweight the scene representation which is not informative for a particular query. More formally, given the total number of clips we are encoding to be N, we modify the equation above as:

$$s(T,V) = \sum_{n=1}^{N} \sum_{i=1}^{K} w_{i,n}(T) s_{i,n}(E_{T_i},E_{V_{i,n}}), \tag{2}$$

$$w_{i,n}(T) = \frac{e^{h(T)^\intercal a_{i,n}}}{\sum_{m=1}^{N} \sum_{j=1}^{K} e^{h(T)^\intercal a_{j,m}}}. \tag{3}$$

Hence instead of learning K scalar weights a_i, $i=1,...,K$ as done in [13] and [38], we learn $K \times N$ scalar weights $a_{i,n}$, $i=1,...,K$, $n=1,...,N$ to allow combination of experts from additional clips.

Dealing with missing streams. We note that these experts might be missing for certain videos, e.g. subtitles are not available for all videos and some videos do not have any detected faces. When expert features are missing, we zero-pad the missing experts and compute the similarity score. This is the standard procedure followed by existing retrieval methods using Mixture of Embedding Experts models [13,38]. The similarity score is calculated only from the available experts by re-normalizing the mixture weights to sum

to one, allowing backpropagation of gradients only to the expert branches that had an input feature. We apply this same principle when dealing with missing video clips in the past, for example if we are training our model with $N=1$ past clips, for a video clip which is right at the start of the movie (has no past), we treat all the experts from the previous clip as missing so that the weights are normalized to focus only on the current clip.

Character Module. The character module computes the similarity between a vector representation of the character IDs mentioned in the query y and a vector representation of the face identities recognised in the clip x. The vector representations are computed as follows: For the query, we search for actor names in the text from the cast list (supplied by the dataset) and create a one-hot vector y the same length as the cast list, where $y_i=1$ if actor i is identified in any face track in the video and $y_i=0$ otherwise. For the face identities acquired in the face recognition pipeline (described earlier), we compare the following three methods: first, we encode a one-hot vector x in a manner similar to the query character encoding. While this can match the presence and absence of characters, it doesn't allow any weighting of characters based on their importance in a clip. Hence inspired by [49], we also propose a second method ("track-frequency normalised"), where x_i is the number of face tracks for identity i. Lastly, in "track length normalised", our vector encodes the total amount of time a character appears in a clip i.e. x_i is the sum of all track lengths for actor i, divided by the total sum of all track lengths in the clip. The performances of the three approaches are displayed and discussed in Table 5 and Section 5 respectively. The character similarity score $s_C=\langle y,x\rangle$ is then modulated by its own scalar mixture weight $w_C(T)$ predicted from $h(T)$ (as is done for the other experts in the model). This similarity score is then added to the similarity score obtained from the other experts to obtain the final similarity score, i.e. $s(T,V)=\sum_{i=1}^{K} w_i(T)s_i(E_{T_i},E_{V_i})+w_C(T)s_C(T,V)$.
Training Loss. As is commonly done for video-text retrieval tasks, we minimise the Bidirectional Max-margin Ranking Loss [50].

5 Experiments

5.1 Experimental Set-up

We train our model for the task of cross-movie and within-movie retrieval. The dataset is split into disjoint training, validation and test sets by movie, so that there are no overlapping movies between the sets. The dataset splits can be seen in Table 3. We report our results on the *test set* using standard retrieval metrics including median rank (lower is better), mean rank (lower is better) and R@K (recall at rank K—higher is better).
Cross-movie Retrieval: For the case of cross-movie retrieval, the metrics are reported over the entire test set of videos, i.e. given a text query, there is a 'gallery' set of 6,581 possible matching videos (Table 3). We report R@1, R@5, R@10, mean and median rank.
Within-movie Retrieval: In order to evaluate the task of within-movie retrieval, we remove all movies that contain less than 5 video clips from the dataset. For each query text, the possible gallery set consists only of the videos in the same movie as the query. In this setting the retrieval metrics are calculated separately for each movie and then averaged over all movies. We report R@1, mean and median rank.

Table 3: Training splits for cross-movie retrieval (left) and within-movie retrieval (right). For within-movie retrieval, we restrict the dataset to movies which have at least 5 video clips in total.

	Cross-Movie				Within-Movie			
	TRAIN	VAL	TEST	TOTAL	TRAIN	VAL	TEST	TOTAL
#Movies	2,551	358	696	3,605	2,469	341	671	3,481
#Video clips	24,047	3,348	6,581	33,976	23,963	3,315	6,581	33,859

5.2 Baselines

The **E2EWS** (End-to-end Weakly Supervised) is a cross-modal retrieval model trained by [51] using weak supervision from a large-scale corpus of (100 million) instructional videos (using speech content as the supervisory signal). We use the video and text encoders without any form of fine-tuning on Condensed Movies, to demonstrate the widely different domain of our dataset.
The **MoEE** (Mixture of Embedded Experts) model proposed by [13] comprises a multi-modal video model in combination with a system of context gates that learn to fuse together different pretrained experts.
The **CE** model [38] similarly learns a cross-modal embedding by fusing together a collection of pretrained experts to form a video encoder, albeit with pairwise relation network sub-architectures. It represents the state-of-the-art on several retrieval benchmarks.
Context Boosting Module: Finally, we report results with the addition of our Context Boosting module to both MoEE and CE. We use the fact that the video clips in our dataset are ordered by the time they appear in the movie, and encode previous and future 'key scenes' in the movie along with every video clip using the CBM. An ablation on the number of clips encoded for context can be found in the supplementary material.
We finally show the results of an ablation study demonstrating the importance of different experts for this task on the task of cross-movie retrieval.

In the next sections, we first describe the implementation details of our models and then discuss quantitative and qualitative results.

5.3 Implementation Details

Expert Features: In order to capture the rich content of a video, we draw on existing powerful representations for a number of different semantic tasks. These are first extracted at a frame-level, then aggregated by taking the mean to produce a single feature vector per modality per video.
RGB object frame-level embeddings of the visual data are generated with an SENet-154 model [52] pretrained on ImageNet for the task of image classification. Frames are extracted at 25 fps, where each frame is resized to 224×224 pixels. Features collected have a dimensionality of 2048.
Motion embeddings are generated using the I3D inception model [53] trained on Kinetics [15], following the procedure described by [53].
Face embeddings for each face track are extracted in three stages: (1) Each frame is passed through a dual shot face detector [54] (trained on the Wider Face dataset [55]) to extract bounding boxes. (2) Each box is then passed through an SENet50 [56] trained on the VGGFace2 dataset [46] for the task of face verification, to extract a facial feature

embedding, which is L2 normalised. (3) A simple tracker is used to connect the bounding boxes temporally within shots into face tracks. Finally the embeddings for each bounding box within a track are average pooled into a single embedding per face track, which is again L2 normalised. The tracker uses a weighted combination of intersection over union and feature similarity (cosine similarity) to link bounding boxes in consecutive frames.
Subtitles are encoded using BERT embeddings [47] averaged across all words.
Scene features of 2208 dimensions are encoded using a DenseNet161 model [57] pretrained on the Places365 dataset [58], applied to 224×224 pixel centre crops of frames extracted at 1fps.
Descriptions are encoded using BERT embeddings, providing contextual word-level features of dimensions $W \times 1024$ where W is the number of tokens. These are concatenated and fed to a NetVLAD layer to produce a feature vector of length of 1024 times the number of NetVLAD clusters for variable length word tokens.
Training details and hyperparameters: All baselines and CBM are implemented with PyTorch [59]. Optimization is performed with Adam [60], using a learning rate of 0.001, and a batch size of 32. The margin hyperparameter m for the bidirectional ranking loss is set to a value of 0.121, the common projection dimension D to 512, and the description NetVLAD clusters to 10. For CBM, we select the number of past and future context videos to be N=3, ablations for hyperparameters and using different amounts of context are given in the supplementary material. Training is stopped when the validation loss stops decreasing.

Table 4: Cross-movie text-video retrieval results on the CMD *test* set of 6,581 video clips, with varying levels of context. Random weights refers to the MoEE model architecture with random initialization. We report Recall@k (higher is better), Median rank and Mean rank (lower is better).

Method	Recall@1	Recall@5	Recall@10	Median Rank	Mean Rank
Random weights	0.0	0.1	0.2	3209	3243.5
E2EWS [51]	0.7	2.2	3.7	1130	1705.5
CE [38]	2.3	7.4	11.8	190	570.0
MoEE [13]	4.7	14.9	22.1	65	285.3
CE + CBM (ours)	3.6	12.0	18.2	103	474.6
MoEE + CBM (ours)	**5.6**	**17.6**	**26.1**	**50**	**243.9**

Table 5: Within-Movie Retrieval results on the CMD test set. All movies with less than 5 video clips are removed. Metrics are computed individually for each movie and then averaged (m-MdR and m-MnR refers to the mean of the median and mean rank obtained for each movie respectively). R@1 denotes recall@1. We show the results of 3 different variations of embeddings obtained from the character module.

Method	m-R@1	m-MdR	m-MnR
Random weights	11.1	5.32	5.32
MoEE	38.9	2.20	2.82
MoEE + Character Module [one-hot]	45.5	1.91	2.60
MoEE + Character Module [track-len norm]	46.2	1.88	2.53
MoEE + Character Module **[track-freq norm]**	**47.2**	**1.85**	**2.49**

Table 6: Expert ablations. The value of different experts in combination with a baseline for text-video retrieval (left) and (right) their cumulative effect (here Prev. denotes the experts used in the previous row). R@k: recall@k, MedR: median rank, MeanR: mean rank

Experts	R@1	R@5	R@10	MedR	MeanR	Experts	R@1	R@5	R@10	MedR	MeanR
Scene	0.8	3.2	5.9	329	776.3	Scene	0.8	3.2	5.9	329	776
Scene+Face	3.7	12.7	19.7	100	443.1	Prev.+Face	3.7	12.7	19.7	100	443.1
Scene+Obj	1.0	4.6	8.0	237	607.8	Prev.+ Obj	3.9	13.1	20.5	79	245.5
Scene+Action	1.9	6.4	10.5	193	575.0	Prev.+ Action	4.0	14.0	20.4	78	233.3
Scene+Speech	2.3	8.3	12.4	165	534.7	Prev.+Speech	5.6	17.7	25.7	50	243.9

Fig. 5: **Qualitative results of the MoEE+CBM model for cross-movie retrieval.** On the left, we provide the input query, and on the right, we show the top 4 video clips retrieved by our model on the CMD *test set*. A single frame for each video clip is shown. The matching clip is highlighted with a green border, while the rest are highlighted in red (best viewed in colour). Note how our model is able to retrieve semantic matches for situations (row 1: male/female on a date), high level abstract concepts (row 2: the words 'stand' and 'truth' are mentioned in the caption and the retrieved samples show a courtroom, men delivering speeches and a policeman's office) and also notions of violence and objects (row 3: scythe).

5.4 Results

Results for cross-movie retrieval can be seen in Table 4. E2EWS performs poorly, illustrating the domain gap between CMD and generic YouTube videos from HowTo100M. Both the CE and MoEE baselines perform much better than random, demonstrating that story-based retrieval is achievable on this dataset. We show that the Contextual Boost module can be effectively used in conjunction with existing video retrieval architectures, improving performance for both CE and MoEE, with the latter being the best performing model. Results for within-movie retrieval can be seen in Table 5. We show that adding in the character module provides a significant boost (almost a 10% increase in Recall@1 compared to the MoEE without the character module), with the best results obtained from normalizing the character embeddings by the track frequency. The value of different experts is assessed in Table 6. Since experts such as subtitles and face are missing for many video clips, we show the performance of individual experts combined with the 'scene' expert features, the expert with the lowest performance that

is consistently available for all clips (as done by [38]). In Table 6, right, we show the cumulative effect of adding in the different experts. The highest boosts are obtained from the face features and the speech features, as expected, since we hypothesize that these are crucial for following human-centric storylines. We show qualitative results for our best cross-movie retrieval model (MoEE + CBM) in Fig. 5.

6 Plot Alignment

A unique aspect of the Condensed Movies Dataset is the story-level captions accompanying the ordered key scenes in the movie. Unlike existing datasets [2] that contain low level visual descriptions of the visual content, our semantic captions capture key plot elements. To illustrate the new kinds of capabilities afforded by this aspect, we align the video descriptions to Wikipedia plot summary sentences using Jumping Dynamic Time Warping [61] of BERT sentence embeddings. This alignment allows us to place each video clip in the global context of the larger plot of the movie. A qualitative example is shown in Fig. 6. Future work will incorporate this global context from movie plots to further improve retrieval performance.

Fig. 6: A sample Wikipedia movie plot summary (left) aligned with an ordered sample of clips and their descriptions (right). The alignment was achieved using Jumping Dynamic Time Warping [61] of sentence-level BERT embeddings, note how the alignment is able to skip a number of peripheral plot sentences.

7 Conclusion

In this work, we introduce a new and challenging *Condensed Movies Dataset* (CMD), containing captioned video clips following succinct and clear storylines in movies. Our dataset consists of long video clips with high level semantic captions, annotated face-tracks, and other movie metadata, and is freely available to the research community. We investigate the task of story-based text retrieval of these clips, and show that modelling past and previous context improves performance. Beside improving retrieval, developing richer models to model longer term temporal context will also allow us to follow the evolution of relationships [34] and higher level semantics in movies, exciting avenues for future work.

Acknowledgements

This work is supported by a Google PhD Fellowship, an EPSRC DTA Studentship, and the EPSRC programme grant Seebibyte EP/M013774/1. We are grateful to Samuel Albanie for his help with feature extraction.

References

1. Tapaswi, M., Bauml, M., Stiefelhagen, R.: Storygraphs: visualizing character interactions as a timeline. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 827–834
2. Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C., Larochelle, H., Courville, A., Schiele, B.: Movie description. International Journal of Computer Vision **123** (2017) 94–120
3. Xiong, Y., Huang, Q., Guo, L., Zhou, H., Zhou, B., Lin, D.: A Graph-Based Framework to Bridge Movies and Synopses. In: The IEEE International Conference on Computer Vision (ICCV). (2019)
4. Vicol, P., Tapaswi, M., Castrejon, L., Fidler, S.: MovieGraphs: Towards Understanding Human-Centric Situations from Videos. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2018)
5. Huang, Q., Xiong, Y., Rao, A., Wang, J., Lin, D.: MovieNet: A Holistic Dataset for Movie Understanding. In: The European Conference on Computer Vision (ECCV). (2020)
6. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. arXiv preprint arXiv:1906.03327 (2019)
7. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: Coin: A large-scale dataset for comprehensive instructional video analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 1207–1216
8. Zhou, L., Xu, C., Corso, J.J.: Towards automatic learning of procedures from web instructional videos. In: Thirty-Second AAAI Conference on Artificial Intelligence. (2018)
9. Ignat, O., Burdick, L., Deng, J., Mihalcea, R.: Identifying Visible Actions in Lifestyle Vlogs. arXiv preprint arXiv:1906.04236 (2019)
10. Fouhey, D.F., Kuo, W.c., Efros, A.A., Malik, J.: From lifestyle vlogs to everyday interactions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 4991–5000
11. Xu, J., Mei, T., Yao, T., Rui, Y.: {MSR-VTT:} A large video description dataset for bridging video and language. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 5288–5296
12. Hendricks, L.A., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: Proceedings of the IEEE international conference on computer vision. (2017) 5803–5812
13. Miech, A., Laptev, I., Sivic, J.: {L}earning a{T}ext-{V}ideo{E}mbedding from {I}ncomplete and{H}eterogeneous{D}ata. In: arXiv. (2018)
14. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: MovieQA: Understanding Stories in Movies through Question-Answering. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
15. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The{Kinetics} Human Action Video Dataset. CoRR **abs/1705.06950** (2017)
16. Gu, C., Sun, C., Ross, D.A., Vondrick, C., Pantofaru, C., Li, Y., Vijayanarasimhan, S., Toderici, G., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: {AVA}: A video dataset of spatio-temporally localized atomic visual actions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 6047–6056
17. Monfort, M., Andonian, A., Zhou, B., Ramakrishnan, K., Bargal, S.A., Yan, Y., Brown, L., Fan, Q., Gutfreund, D., Vondrick, C., Others: Moments in time dataset: one million videos for event understanding. IEEE transactions on pattern analysis and machine intelligence (2019)
18. Sener, O., Zamir, A.R., Savarese, S., Saxena, A.: Unsupervised semantic parsing of video collections. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 4480–4488

19. Alayrac, J.B., Bojanowski, P., Agrawal, N., Sivic, J., Laptev, I., Lacoste-Julien, S.: Unsupervised learning from narrated instruction videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2016) 4575–4583
20. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. arXiv preprint arXiv:1904.01766 (2019)
21. Everingham, M., Sivic, J., Zisserman, A.: "Hello! My name is... Buffy" – automatic naming of characters in TV video. In: Proc. BMVC. (2006)
22. Naim, I., Al Mamun, A., Song, Y.C., Luo, J., Kautz, H., Gildea, D.: Aligning movies with scripts by exploiting temporal ordering constraints. In: 2016 23rd International Conference on Pattern Recognition (ICPR), IEEE (2016) 1786–1791
23. Cour, T., Sapp, B., Jordan, C., Taskar, B.: Learning from ambiguously labeled images. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2009) 919–926
24. Sivic, J., Everingham, M., Zisserman, A.: "Who are you?" – learning person specific classifiers from video. In: Proc. CVPR. (2009)
25. Tapaswi, M., Bäuml, M., Stiefelhagen, R.: "Knock! Knock! Who is it?" probabilistic person identification in TV-series. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2012) 2658–2665
26. Huang, Q., Yang, L., Huang, H., Wu, T., Lin, D.: Caption-Supervised Face Recognition: Training a State-of-the-Art Face Model without Manual Annotation. In: The European Conference on Computer Vision (ECCV). (2020)
27. Bojanowski, P., Bach, F., Laptev, I., Ponce, J., Schmid, C., Sivic, J.: Finding actors and actions in movies. In: Proceedings of the IEEE international conference on computer vision. (2013) 2280–2287
28. Duchenne, O., Laptev, I., Sivic, J., Bach, F., Ponce, J.: Automatic annotation of human actions in video. In: 2009 IEEE 12th International Conference on Computer Vision, IEEE (2009) 1491–1498
29. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. (2008)
30. Marszałek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR 2009-IEEE Conference on Computer Vision & Pattern Recognition, IEEE Computer Society (2009) 2929–2936
31. Nagrani, A., Sun, C., Ross, D., Sukthankar, R., Schmid, C., Zisserman, A.: Speech2Action: Cross-Modal Supervision for Action Recognition. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
32. Ercolessi, P., Bredin, H., Sénac, C.: StoViz: story visualization of TV series. In: Proceedings of the 20th ACM international conference on Multimedia. (2012) 1329–1330
33. Rao, A., Xu, L., Xiong, Y., Xu, G., Huang, Q., Zhou, B., Lin, D.: A Local-to-Global Approach to Multi-modal Movie Scene Segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 10146–10155
34. Kukleva, A., Tapaswi, M., Laptev, I.: Learning Interactions and Relationships between Movie Characters. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'20). (2020)
35. Jasani, B., Girdhar, R., Ramanan, D.: Are we asking the right questions in{MovieQA?}. In: ICCVW. (2019)
36. Nagrani, A., Zisserman, A.: From benedict cumberbatch to sherlock holmes: Character identification in tv series without a script. In: Proc. BMVC. (2017)
37. Lei, J., Yu, L., Berg, T.L., Bansal, M.: TVQA+: Spatio-Temporal Grounding for Video Question Answering. In: Tech Report, arXiv. (2019)
38. Liu, Y., Albanie, S., Nagrani, A., Zisserman, A.: Use what you have: Video retrieval using representations from collaborative experts. In: Proc. BMVC. (2019)
39. Tapaswi, M., Bäuml, M., Stiefelhagen, R.: Aligning plot synopses to videos for story-based retrieval. International Journal of Multimedia Information Retrieval **4** (2015) 3–16
40. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Carlos Niebles, J.: Dense-captioning events in videos. In: Proceedings of the IEEE international conference on computer vision. (2017) 706–715

41. Wu, C.Y., Feichtenhofer, C., Fan, H., He, K., Krahenbuhl, P., Girshick, R.: Long-Term Feature Banks for Detailed Video Understanding. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2019)
42. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: ActivityNet: A large-scale video benchmark for human activity understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 961–970
43. Nagrani, A., Chung, J.S., Xie, W., Zisserman, A.: Voxceleb: Large-scale speaker verification in the wild. Computer Speech and Language (2019)
44. Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C., Larochelle, H., Courville, A., Schiele, B.: Movie Description. International Journal of Computer Vision (2017)
45. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: VGGFace2: A dataset for recognising faces across pose and age. In: Proc. Int. Conf. Autom. Face and Gesture Recog. (2018)
46. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
47. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 5297–5307
48. Tapaswi, M., Bäuml, M., Stiefelhagen, R.: Book2Movie: Aligning Video scenes with Book chapters. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2015)
49. Socher, R., Karpathy, A., Le, Q.V., Manning, C.D., Ng, A.Y.: Grounded compositional semantics for finding and describing images with sentences. Transactions of the Association for Computational Linguistics **2** (2014) 207–218
50. Miech, A., Alayrac, J.B., Smaira, L., Laptev, I., Sivic, J., Zisserman, A.: End-to-End Learning of Visual Representations From Uncurated Instructional Videos. In: The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). (2020)
51. Hu, J., Shen, L., Albanie, S., Sun, G., Wu, E.: Squeeze-and-Excitation Networks. IEEE transactions on pattern analysis and machine intelligence (2019)
52. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the{Kinetics} dataset. In: proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2017) 6299–6308
53. Li, J., Wang, Y., Wang, C., Tai, Y., Qian, J., Yang, J., Wang, C., Li, J., Huang, F.: DSFD: dual shot face detector. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2019) 5060–5069
54. Yang, S., Luo, P., Loy, C.C., Tang, X.: WIDER FACE: A Face Detection Benchmark. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016)
55. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proc. CVPR. (2018)
56. Iandola, F., Moskewicz, M., Karayev, S., Girshick, R., Darrell, T., Keutzer, K.: Densenet: Implementing efficient convnet descriptor pyramids. arXiv preprint arXiv:1404.1869 (2014)
57. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE transactions on pattern analysis and machine intelligence **40** (2017) 1452–1464
58. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in pytorch. (2017)
59. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. CoRR **abs/1412.6980** (2014)
60. Feng, L., Zhao, X., Liu, Y., Yao, Y., Jin, B.: A similarity measure of Jumping Dynamic Time Warping. In: 2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery. Volume 4. (2010) 1677–1681