Editing like Humans: A Contextual, Multimodal Framework for Automated Video Editing

Sharath Koorathota[1,2,*] Patrick Adelman[2,*] Kelly Cotton[3] Paul Sajda[1]

[1]Dept. of Biomedical Engineering, Columbia University; [2]Fovea Inc.;
[3]Dept. of Psychology, The Graduate Center, CUNY

sk4172@columbia.edu, patrick@foveainsights.com, kcotton@gradcenter.cuny.edu, psajda@columbia.edu

Abstract

We propose an automated video editing model, which we term contextual and multimodal video editing (CMVE). The model leverages visual and textual metadata describing videos, integrating essential information from both modalities, and uses a learned editing style from a single example video to coherently combine clips. The editing model is useful for tasks such as generating news clip montages and highlight reels given a text query that describes the video storyline. The model exploits the perceptual similarity between video frames, objects in videos and text descriptions to emulate coherent video editing. Amazon Mechanical Turk participants made judgements comparing CMVE to expert human editing. Experimental results showed no significant difference in the CMVE vs human edited video in terms of matching the text query and the level of interest each generates, suggesting CMVE is able to effectively integrate semantic information across visual and textual modalities and create perceptually coherent quality videos typical of human video editors. We publicly release an online demonstration of our method.

1. Introduction

The explosion of consumer and professional videography has brought with it a profound challenge to both computer vision and language processing communities. Existing approaches to editing videos from large amounts of raw footage involve the use of editing software to organize footage, metadata associated with footage, and human effort [10]. However, editing video with existing video editing systems is a tedious and time-consuming effort, and requires an editor with significant editing skills and creativity [21].

Video editing is often a field-specific creative effort. For example, news content editing may rely on compilations of source content while film editing is typically more character- or story-reliant [28]. Once provided with a video storyline (i.e., text query), which may include key characters, actions and the causal relationships [14], and raw footage, the primary set of tasks video editors complete include: aggregation of relevant footage, search of footage for query-relevant clips, compilation of clips into a coherent video and iterative correction of video (e.g., color correction, audio mixing) [22].

Recent efforts to automate video editing have focused on improving individual aspects of the editing process. For example, deep learning advancements in entity and object recognition paired with semantic descriptions of visual scenes have improved indexing and speed of retrieval of task-relevant content from video collections [26, 30]. Visual style transfer algorithms have been used to describe and enforce perceptual similarity in images and video [32]. Natural language embeddings have been useful in describing similarity of metadata [29].

Here we propose an integrated approach[1] to automate editing which utilizes deep learning advancements in object detection, natural language processing (NLP), and perceptual similarity to describe video content and models creative decision parameters learned through example videos. Specifically, in this paper we:

- present a technique to construct a video from a set of source clips through parsing a text query for relevant entities and objects,
- describe a simple method to model video editing style given a sample video, enforcing perceptual similarity, object tracking, and the presence of important entities

*These authors contributed equally

[1]A public, online demonstration of our proposed model can be found at: http://cmve.foveainsights.com

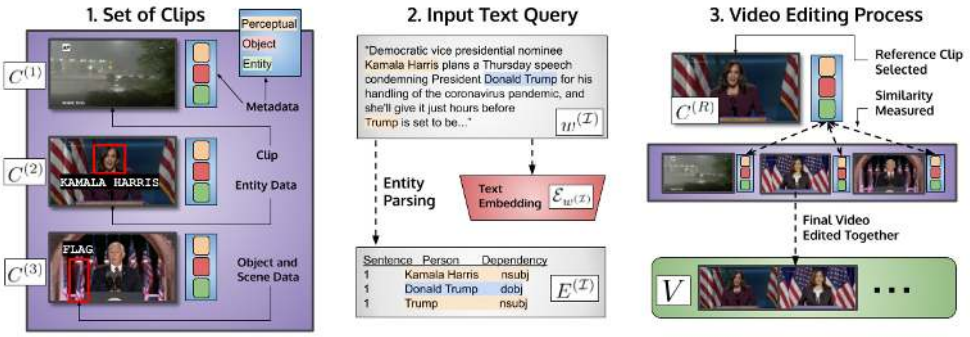

Figure 1. **System Overview** - 1. Videos are processed to extract metadata, including visual, object, and entity data. 2. An input text query is parsed for entity information and embedded using a universal sentence encoder. 3. Videos are split into clips, ranked by text embedding and entity presence, and are compared to a reference clip to edit the final video, using a set of input film rules.

while allowing for creative, field-specific parameterization,

- demonstrate video editing applications using footage from a news content collection to edit video specific to short text queries,
- evaluate our model through human judgements of CMVE- and human professionally-edited videos in participants recruited on Amazon MTurk.

2. Related Work

2.1. Multimodal Reasoning

While past work used the spatial-temporal relation of objects in video for retrieval, allowing discrimination power about different video criteria [30], recent research related to multimodal video retrieval has focused on improving natural language descriptors [23] or extracting key events from videos [31, 12]. Universal embeddings to measure semantic similarity between two images have also been employed successfully within domains of image processing with potential to extend to video, but require large amounts of data from multiple domains [9]. Similarly, deep learning approaches with collaborative interfaces between textual and visual data in scenes have yielded encouraging results for estimating similarity between images and sentences through focusing on shared semantics [8].

2.2. Scene Coherency

Video editing first requires segmentation of videos into scenes, followed by modeling the relationships between scenes for a coherent video edit. Recent work in the area has used shot similarity graphs to represent shots and their boundaries using color and motion information [24]. Scene graphs, through relating objects as nodes and the pairwise relationships as edges, have yielded interpretable and accurate results in real-world images [25]. While using objects in video may provide an encouraging path to determine relevancy of scenes while editing, automated methods have previously focused on domain specific video clips and used object detectors with limited vocabularies [3].

2.3. Modeling Editing Workflows

Video editing often requires complex workflows. In the first stage, existing footage requires extensive searching, bookmarking and reviewing to find key moments leveraging metadata [11]. In the second stage, specific timepoints in a scene are identified for cutting clips together seamlessly. Achieving video coherence is typically the most tedious aspect of editing, requiring planning of how source clips can be aligned effectively and how these clips will be chunked together in the output video [21, 15]. Compositing, sound mixing, overlays and other visual effects are planned at this stage and relate to overall video coherency [22]. Finally, a review phase of the final video is used to inform editors whether the editing goal is achieved. Depending on project deadline, collaborative input, and size of the video collection, this process is generally iterative.

Recent efforts to automate video editing have focused on addressing the navigation phase through improving the quality of search, video metadata, or interfaces, or improving the speed of creating visual effects and workflows during editing. To our knowledge, we are the first to propose a pipeline for automated editing where the goal is driven by a planned storyline of the output video.

3. Video Model

Broadly, our video model $\mathcal{M}_V$ outputs a video V and takes input $\mathcal{I}$, a collection of existing videos $\mathcal{D}$, and a reference video $V^{(R)}$:

$$V = \mathcal{M}_{\mathcal{V}}\left(\mathcal{I}, \mathcal{D}, V^{(R)}\right) \tag{1}$$

where the user input $\mathcal{I} = \{w^{(\mathcal{I})}, \mathcal{I}_f, \mathcal{I}_c\}$ is comprised of

a text query $w^{(\mathcal{I})}$ that describes the desired output video, a set of film rules $\mathcal{I}_f$, a set of input constraints $\mathcal{I}_c$. The collection of videos $\mathcal{D}$ is indexed by i, where each element $V^{(i)}$ is pair $(v^{(i)}, w^{(i)})$, a video $v^{(i)}$ and associated text description $w^{(i)}$.

To output a video that is edited comparably to a human, our model must parse both the input text description of the desired output video and the collection of videos that are available to use. We use a combination of sentence encoding, sentence segmentation, and named entity recognition [16] to identify what a text is both generally and specifically about, and to create an outline for the final video for our model to fill in. For video processing, CMVE uses traditional video metadata for filtering and deep learning networks to segment videos into individual clips, identify objects, and create perceptual profiles of each clip.

Given the textual and visual information passed into our model, the model filters the collection for relevant videos, subdivides the videos into clips, and uses both textual and visual context to select the most appropriate reference clip for a given sentence. The model then produces a linearly weighted score for every remaining clip based on its corresponding visual information in comparison to the reference clip. The ranking of clips becomes the final order that is edited together.

4. Algorithm

The aim of our model is to take an input text and edit an appropriate video V to match that text, pulling from a source collection of videos. The collection may be large and contain videos that are unrelated or undesirable for the final video. Initially, our algorithm proceeds by comparing the input text to video metadata in order to narrow down the large collection of videos to those that are related to the input text. It then divides those videos into a set of clips, analyzes a reference video for known entities and objects, and computes pairwise similarity weights with non-reference clips. Finally, CMVE uses the learned similarity weights to compute comparison scores for clips, selects the highest ranking clips that match the input text and outputs video V. Selected notation can be referenced in Supplementary Table 4.

4.1. Ranking video descriptions

First, we consider global input constraints $\mathcal{I}_c$, and filter the videos in $\mathcal{D}$ to improve search quality:

$$\mathcal{M}_K : \left\{ \mathbb{1}\left(w^{(\mathcal{I})}, w^{(i)}, \mathcal{I}_c\right) \right\} \mapsto T_1, \forall\, V^{(i)} \in \mathcal{D} \qquad (2)$$

where $\mathcal{M}_K$ in our case uses an indicator function to filter videos that meet a minimum duration threshold and whose

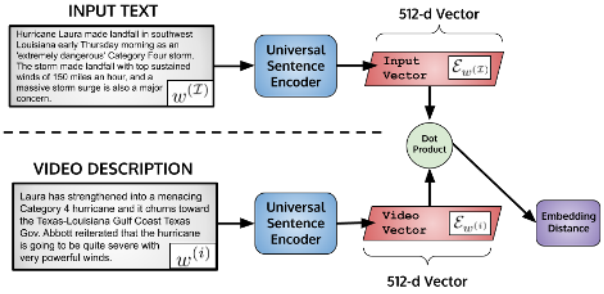

Figure 2. **Text Embeddings** - For the input text and the text descriptions of a video, we calculate a $512 - d$ vector for each using the universal sentence encoder. Then we calculate the inner product of each video's text description vector against the input text vector.

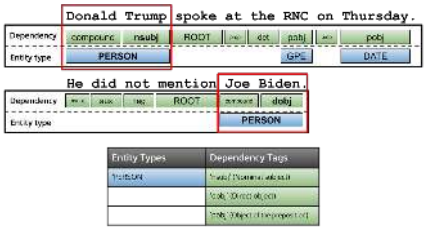

Figure 3. **Named Entity Recognition** - For each input script, the named entity extraction model searches for entities of a given type ('PERSON') and then searches those entities for ones that match given dependency tags ('nsubj', 'dobj', 'pobj'). Specifications follow ClearNLP guidelines [27].

associated video descriptions $w^{(i)}$ contain the presence of at least one word from $w^{(\mathcal{I})}$.

Next, we use a sentence encoding model, which generates a 512-dimensional embedding of text of varying lengths [5], to rank $n \in \mathcal{I}_c$ most relevant videos descriptions through:

$$\mathcal{M}_E : \max_n \{\mathcal{E}_{w^{(\mathcal{I})}} \cdot \mathcal{E}_{w^{(i)}}\} \mapsto T_2, \forall w^{(i)} \in V^{(i)} \in T_1 \quad (3)$$

where each human-generated text description $w^{(i)} \in V^{(i)}$ in $\mathcal{D}$ is compared pairwise with the text query $w^{(\mathcal{I})}$. Embeddings $\mathcal{E}_{w^{(\mathcal{I})}}, \mathcal{E}_{w^{(i)}} \in \mathbb{R}^{512}$ are computed using a universal sentence encoder [5]. The distance between the two embeddings is computed as a simple inner product (Fig. 2). The set of n most related video description embeddings $\mathcal{E}_{w^{(i)}}$ are used to define an ordered list of top videos T_2 that correspond to these descriptions.

4.2. Sentence parsing and clip segmentation

Sentence Parsing. We next define each sentence j of input text $w^{(\mathcal{I})}$ to represent a distinct and complete visual thought in the output video, using past work showing sen-

tences may be the minimum meaningful unit in defining intervals of video [29] and in boundaries during natural conversation [19]. Using a syntactic parser trained on large amounts of language data [16], we extract (Fig. 3) for each visual thought $w_j^{(\mathcal{I})} \in w^{(\mathcal{I})}$:

1. expected duration $d_j^{(\mathcal{I})}$ of video output for sentence j, using an average speaking duration informed by observational data [17],
2. set of named entities $E_j^{(\mathcal{I})}$ (e.g. 'John Doe')
3. set of relevant dependency tags [27], e.g. nominal subjects, direct objects, and objects of the preposition, with respect to sentence j, P_j.

Clip Segmentation. We segment our video data into clips with a clip segmentation model $\mathcal{M}_C$ using function $\mathbb{C}$:

$$\mathcal{M}_C : \left\{ \mathbb{C}\left(V^{(i)}, \mathcal{I}_c\right) \right\} \mapsto N, \; \forall \, V^{(i)} \in T_2 \qquad (4)$$

where the output N is a set of m clips $\{C^{(1)}, \ldots, C^{(m)}\}$ grouped by ranked, parent videos $V^{(i)}$ in T_2. Each clip is a visually-uninterrupted segment from a video, often referred to as a shot in filmmaking [24]. $\mathbb{C}$ uses a trained model [2] to detect shots, and filters for shots meeting a confidence level and minimum duration constraints specified in $\mathcal{I}_c$. This process is outlined in Supplementary Algorithm 2.

4.3. Clip metadata

For each clip $C^{(m)}$ in N we derive and process new associated metadata:

1. a clip thumbnail $b^{(m)}$,
2. a set of bounded objects $O^{(m)}$,
3. a set of unbounded objects $U^{(m)}$,
4. a set of recognizable entities $E^{(m)}$,

where $b^{(m)}$ is calculated as the midpoint frame of the clip, and serves as an efficient, visual representation of the clip for similarity analysis. Bounded object data $O^{(m)}$ included only tangible objects, for example labels such as "human" or "podium." Unbounded data $U^{(m)}$ were scene descriptors that were unable to be bounded but still recognizable, such as "symbol" or "speech." Entities $E^{(m)}$ are defined as characters that are recognizable by a facial detection and recognition framework by name, e.g. celebrities, who are treated separately for the purpose of learning semantic structure from clips. In our experiments, object and entity detection data were generated from the same object detection framework $\mathcal{M}_O$ [1].

4.4. Training from a single sample

Given a single training sample comprising of a video $V^{(T)}$ and text query $w^{(T)}$, we segment the video into clips and learn the importance of perceptual and object similarity within $V^{(T)}$.

We first segment the video into clips C using the process described in Eq. 4. Given by the importance of the first shot in "setting up a scene," [7, 18, 13] we label the first clip as the reference clip $C^{(R)} = C^{(1)}$ and calculate pairwise scores $s_{\mathbf{p}}^{(m)}$, $s_{\mathbf{b}}^{(m)}$, $s_{\mathbf{u}}^{(m)}$, representing the perceptual similarity, the bounded object similarity, and the unbounded object similarity of each remaining clip in $\{C^{(2)}, \ldots, C^{(m)}\}$ with respect to $C^{(R)}$ for m clips derived from $V^{(T)}$ (Fig. 4). Selected, early comparisons of news videos and non-news videos showed high specificity in classifications across perceptual and object similarity dimensions (Supplementary Fig. 10).

4.4.1 Perceptual similarity

The relative perceptual similarity score $s_{\mathbf{p}}^{(m)}$ for a clip to a reference clip is calculated first by using layer activations from the AlexNet architecture [20]:

$$d(b^{(m)}, b^{(R)}) = \sum_l \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot \left(\hat{y}_{hw}^{(l)} - \hat{y}_{0hw}^{(l)} \right) \right\|_2^2 \qquad (5)$$

where the distance d is calculated for a pair of clips using the compared clip's thumbnail $b^{(m)}$ and the reference clip thumbnail $b^{(R)}$, using scaled activations channel-wise by vector w_l, and the layer activations $\hat{y}_{hw}^l$ and $\hat{y}_{0hw}^l$ at layer l given dimensions H_l and W_l. We then map the distance to a perceptual similarity score for the clips using a small network:

$$s_{\mathbf{p}}^{(m)} = 1 - \mathcal{G}\left(d(b^{(m)}, b^{(R)}) \right) \qquad (6)$$

where $\mathcal{G}$ is a 32-channel, fully-connected network that maps the distance between two thumbnails to a judgement of perceptual similarity $\in (0, 1)$. We used the lin architecture and training methods described by Zhang et al [32] to learn this mapping.

Eq. 6 maps layer activations to human judgements of differences between images, allowing CMVE to model perceptually-driven editing decisions for clip selection.

4.4.2 Object similarity

For the reference clip and each compared clip we have a list of bounded objects $O^{(R)}$ and $O^{(m)}$ respectively. We then consider $O^{(R)\prime}$ and $O^{(m)\prime}$, vectors that represent the counts of each unique object that exist in both clips.

The relative bounded object similarity score $s_{\mathbf{b}}^{(m)}$ for a clip to a reference clip is calculated using lists of bounded objects:

$$s_{\mathbf{b}}^{(m)} = \frac{\sum_{n=1}^{N} |O_n^{(m)\prime}||O_n^{(R)\prime}|}{\sqrt{\sum_{n=1}^{N} |O_n^{(m)\prime}|^2}\sqrt{\sum_{n=1}^{N} |O_n^{(R)\prime}|^2}} \tag{7}$$

where N is total count of objects from $O^{(m)} \cup O^{(R)}$, and we use the cosine similarity between the bounded object count vectors $O^{(m)\prime}$ for the compared clip and bounded objects $O^{(R)\prime}$ for the reference clip as the score.

We use same process to calculate an unbounded object similarity score $s_{\mathbf{u}}^{(m)}$:

$$s_{\mathbf{u}}^{(m)} = \frac{\sum_{i=1}^{Y} |U_i^{(m)\prime}||U_i^{(R)\prime}|}{\sqrt{\sum_{i=1}^{Y} |U_i^{(m)\prime}|^2}\sqrt{\sum_{i=1}^{Y} |U_i^{(R)\prime}|^2}} \tag{8}$$

where Y is the total count of objects from $U^{(m)} \cup U^{(R)}$, and the similarity between the between the unbounded object count vectors $U^{(m)\prime}$ for the compared clip and unbounded objects $U^{(R)\prime}$ for the reference clip is computed.

4.4.3 Entity Overlap

A notable challenge in reasoning from and describing video scenes which affects video editing is in understanding how major characters or entities relate to each other in a scene [25, 12]. This is a particular challenge when text queries used to search video can include novel entities that are not recognized by an object detection model. To account for this mismatch, and learn the semantic style of a training query $w^{(T)}$ with j sentences. For each sentence $w_j^{(T)}$, we first identify named entities $E_j^{(T)}$ for that sentence as described in in Section 4.2. We then consider for each clip $C^{(m)}$ the entity metadata $E^{(m)}$ for that clip. We calculate an entity overlap weight vector $\mathbf{e_w}$ over all training sentences as:

$$e_{\mathbf{c}} =< \left(\frac{\sum_j |E_j^{(T)} \cap E^{(m)}|}{j} \right)_p >, \forall p \in P_j \tag{9}$$

$$\mathbf{e_w} =< \frac{\exp(e_{\mathbf{c}})}{\sum \exp(e_{\mathbf{c}})} > \tag{10}$$

where we count the detected entities in a clip that overlap with the input text query $w^{(T)}$. Thus each index of $e_{\mathbf{c}}$ contains an average count of entities in the clip $C^{(m)}$ that match that dependency tag P_j across all sentences of $w^{(T)}$. The entity overlap $\mathbf{e_w}$ given a clip $C^{(m)}$ and a $w_j^{(T)}$ is the softmax score vector of counts by dependency tag, determined by size P_j. This is computed as an average across all clips in the training video, and defines the overlap of known entities, by dependency tag, between the text query and known objects in $\mathcal{M}_O$ across all reference clips.

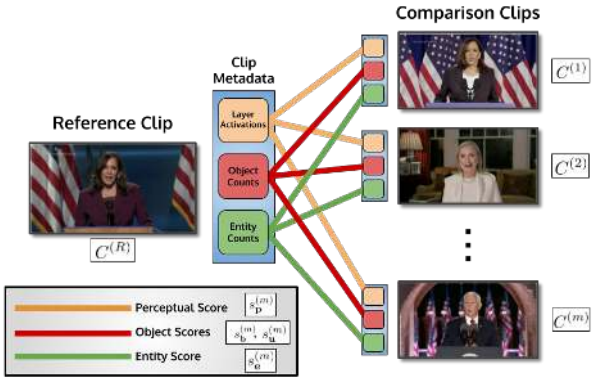

Figure 4. **Clip Scores** - For each clip in T_2, the perceptual, bounded object, unbounded object, and entity scores, with regards to the reference clip, are calculated and compiled into a score vector.

In this way, we use dependency tags to semantically link objects that *should be* described in the output video, which we source from $w^{(T)}$, with *known objects* in an object detection framework $\mathcal{M}_O$.

Finally, we arrive at the entity score $\mathbf{s_e}$ for a clip $C^{(m)}$ by weighting dependency tag-grouped counts by $\mathbf{e_w}$:

$$s_{\mathbf{e}}^{(m)} = \mathbf{e_w} \cdot < \left(|E_j^{(T)} \cap E^{(m)}| \right)_p >, \forall p \in P_j \tag{11}$$

for j, the parent sentence from a text query that corresponds to a clip m. For each clip, the score is calculated using relevant dependency tag weights, captured in $\mathbf{e_w}$, for each tag $p \in P_j$.

4.4.4 Score vector

We calculate a score vector by transforming all calculated scores, after averaging across non-reference clips, through the softmax:

$$\mathbf{s_w} =< \frac{\exp(s)}{\sum_s \exp(s)} >, \forall s \in s_{\mathbf{p}}^{(m)}, s_{\mathbf{b}}^{(m)}, s_{\mathbf{u}}^{(m)}, s_{\mathbf{e}}^{(m)} \tag{12}$$

We use this weight vector of scores $\mathbf{s_w}$ as a reference in order to weight the importance of clip metadata when editing a video from a novel, unseen input query.

4.5. Per-sentence clip rankings

Given an unseen input query $w^{(\mathcal{I})}$, list of ordered clips T_2, an entity profile vector $\mathbf{e_w}$, and a learned entity score

vector $\mathbf{s_w}$, we define the reference clip as the clip that has the maximum entity score $s_\mathbf{e}$ (as computed in Eq. 11)

$$C^{(R)} = \arg\max_{C} \left(s_\mathbf{e}^{(m)} \right), \forall\, C^{(m)} \in T_2 \tag{13}$$

where the reference clip $C_j^{(R)}$ is selected for a given sentence $w_j^{(\mathcal{I})}$.

Next, for each sentence j, we rank all other clips $\{C^{(2)}, \ldots, C^{(m)}\}$ in T_2 by their weighted perceptual, object and entity similarity, using the learned score vector from Eq. 12 and the scores calculated in Eqs. 5-11:

$$s^{(m)} = \mathbf{s_w}^\bullet < s_\mathbf{p}^{(m)}, s_\mathbf{b}^{(m)}, s_\mathbf{u}^{(m)}, s_\mathbf{e}^{(m)} > \tag{14}$$

$$\text{rank}\{s^{(m)}\} \mapsto T_3, \forall\, C^{(m)} \in T_2, C^{(m)} \neq C^{(R)} \tag{15}$$

where $s^{(m)}$ is the weighted score of clip m, T_3 is a set of clips ranked by weighted similarity score with respect to the reference clip $C^{(R)}$, in descending order. Because $\mathbf{s_w}$ is learned from a reference video, weights learned in this framework capture the relative importance of perceptual, object and entity similarity, defining a style for the edit. We define a per-sentence ordered list of clips as N_C:

$$N_C = \{C^{(R)}, C^{(1)}, ..., C^{(i)}\},\ \forall\, C \in T_3 \tag{16}$$

where i is the number of clips in T_3.

4.6. Video editing

The final CMVE video V is edited using N_C, for each sentence:

$$\mathcal{M}_S(\mathcal{I}_f, N_C, d_j^{(\mathcal{I})}) : Y^{(j)} = \{N_0^{(j)} \oplus N_1^{(j)} ... \oplus N_m^{(j)}\} \tag{17}$$

$$V = \{Y^{(1)} \cup \cdots \cup Y^{(j)}\} \tag{18}$$

where $\mathcal{M}_S$ is a process which stitches together a sub-video $Y^{(j)}$ for each sentence j of $w_j^{(\mathcal{I})}$. We use clip transitions $\oplus$ specified in $\mathcal{I}_f$ (e.g. fade-in effect), and a duration to threshold the number of clips per sentence, computed by expected speaking duration of the sentence $d_j^{(\mathcal{I})}$.

The final video V is then stitched together as the union of each sub-video $Y^{(j)}$. In our experiments, we defined a cutoff for the number of clips m per sentence in $\mathcal{I}_f$, but this may also be learned from a reference clip.

5. Experiments

5.1. Experimental Settings

Dataset. We applied our framework on a news compilation (NC) video dataset from the Associated Press Newsroom [4]. Each video highlights the top stories of the day in a short, 1-2 minute compilation video, and contains a voice-over narration. We assessed videos released 08-18-2020 to 08-27-2020, chosen randomly across a variety of topics relevant to the US news market: on the COVID-19 pandemic ("Covid"), Hurricane Laura ("Hurricane"), protests on racial injustice ("Protests"), the United States Postal Service ("USPS"), the Republican National Convention ("RNC") and the Democratic National Convention ("DNC"). We chose 2 videos for each topic for a total of 12 videos, each containing a text script $w^{(\mathcal{I})}$ with two sentences. Each video was edited by a human editor, ranged in duration from 11-28 seconds, and contained 2-5 individual clips. While we did not have access to all of the source clips available to the human editor, $\mathcal{D}$ is comprised of $1,493$ news clips of varying style and source content. Video details are provided in Supplementary Table 1.

Model Training. First, we define the reference clip for each training sample (i.e. human-edited video) as the first clip after segmenting using $\mathcal{M}_C$ [2]. Based on past work suggesting the importance of using subject and object of an image in connection to video semantics [31], we fix P_j to be a vector of nominal subject, direct object and object of the preposition for sentences j and $j-1$, $P_j =< \textit{nsubj}_j, \textit{dobj}_j, \textit{pobj}_j, \textit{nsubj}_{j-1}, \textit{dobj}_{j-1}, \textit{pobj}_{j-1} >$ [27]. For the first sentence, values corresponding to the previous sentence are set to 0. We calculate $s_\mathbf{p}^{(m)}$, $s_\mathbf{b}^{(m)}$, $s_\mathbf{u}^{(m)}$ and $s_\mathbf{e}^{(m)}$ for each non-reference clip using Eqs. 5-12. Due to the small training sample dataset size, we average the weight vector calculated using Eq 12 first across all clips within a training sample video, and then across all training sample videos. This learned $\mathbf{s_w}$ is used to edit videos from a larger collection of videos $\mathcal{D}$.

The number of T_1 collection videos considered for each training sample ranged from 39-182, and the number of clips segmented by $\mathcal{M}_C$ ranged from 140-242. Each video in T_1 had a human-annotated description of approximately 5 sentences in length that described the content of the video, which we use to rank through Eq 3, using training sample video scripts as $w^{(\mathcal{I})}$. Clip metadata was generated using AWS Rekognition [1], resulting in $b^{(m)}$, $O^{(m)}$, $U^{(m)}$, and $E^{(m)}$. For each sentence $w_j^{(\mathcal{I})}$ of the training sample query, we define reference clips $C_j^{(R)}$ for sentence j using Eq. 13. Next, for all other clips x from T_1 we calculate $s_\mathbf{p}^{(x)}$, $s_\mathbf{b}^{(x)}$, $s_\mathbf{u}^{(x)}$ and $s_\mathbf{e}^{(x)}$ as described in Eqs. 5-12 and calculate the dot product with $\mathbf{s_w}$ to compute a weighted score using Eq 14, rank clips using Eq. 15 and arrive at the CMVE edit using Eq. 18.

Study. We conducted a human subject study using Amazon MTurk to compare CMVE videos to those edited by human editors in the above training samples. We recruited $N = 80$ study participants (Mean age = 42.8, 50 males, 23 females, and 7 unknown). Participants were chosen from a

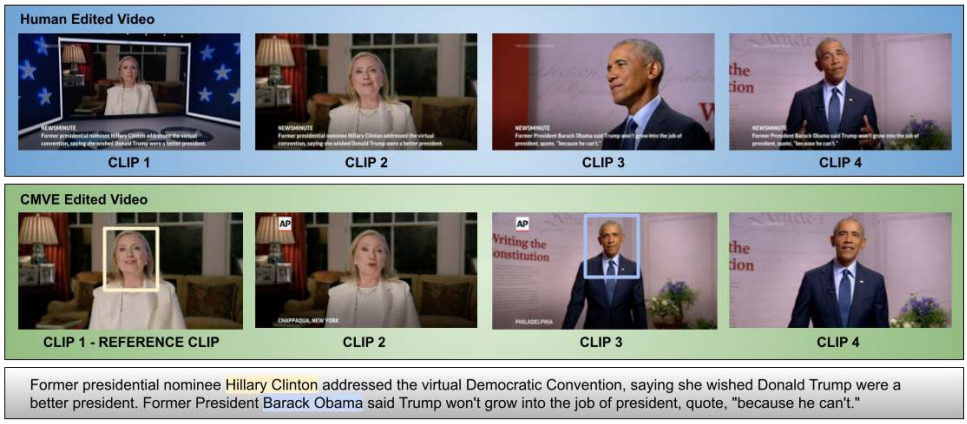

Figure 5. Example of a human edited video side by side with a CMVE edited video. The text prompt is the voice over script for the video, and the input text $w^{(\mathcal{I})}$ to CMVE. Highlighted names "Hillary Clinton" and "Barack Obama" are known entities in $E_j^{(\mathcal{I})}$ are matched to clips that are edited into the video.

pool of US residents. Each participant was informed that they would be shown a piece of text and a video meant to correspond to that text. Participants were surveyed after each video about the quality of the video and their viewing experience.

In total, we compared 24 videos: two edits (CVME and "Human") of 12 videos, corresponding to the topics chosen for the trained model, with two videos belonging to each topic. Because we used a limited set of clips for the CMVE-edited videos, not all footage from the ground truth edits were available for automated editing, posing a challenge for CMVE. Each participant was shown one video from each of the 6 topics, and were not informed as to whether the videos were human- or CMVE-edited.

Eight questions were displayed after each video assessing the overall quality of video and audio, as well as specific aspects of the viewing experience. The primary evaluation metric was whether participants rated the quality of CMVE differently than human edited video. Participants were required to answer all questions on all videos in order to complete the study. A brief demographic survey was collected at the end of the study, including questions about each participants weekly print and video news viewing habits. Participant demographics can be found in Supplementary Table 2.

5.2. Results

We arrived at the following, learned, $\mathbf{e_w}$ and $\mathbf{s_w}$ which specify an interpretable, "editing style" from reference videos:

$$\mathbf{e_w} =< 0.60, 0.11, 0.06, 0.10, 0.06, 0.06 > \quad (19)$$

$$\mathbf{s_w} =< 0.22, 0.31, 0.28, 0.19 > \quad (20)$$

where $\mathbf{e_w}$ suggests reliance on the current sentence, overlapping, nominal subject entity in the NC dataset. Furthermore, $\mathbf{s_w}$ reinforces a relatively greater reliance on object similarity, over perceptual similarity, for a news editing style. Because news video is often sourced from multiple clips and compilation-style edits are ubiquitous [6], objects may be more important in determining style for video coherency than perceptual similarity of color profiles.

The primary evaluation metric was whether participants rated the quality of CMVE differently than human edited ("Human") video. Figure 6 shows the ratings in our primary outcomes of interest. There was, on average, $< 5\%$ rating difference between CMVE and "Human" ratings in 6 topics of news videos, averaged across two videos per topic. The difference was largest in the "dnc" topic, and lowest for "hurricane."

Two-way repeated measure analyses of variance (ANOVAs) were conducted to evaluate the effect of editor and video topic on quality ratings. We found a significant main effect of editor for question 3 ("I would expect to see this on a news channel"), $F(1, 453) = 4.906$, $p = .027$, and question 6 ("The video was informative"), $F(1, 454) = 4.840$, $p = .028$. Additionally, there was a significant interaction between editor and video for question 1 ("Please rate the overall quality of the video"), $F(5, 456) = 2.652$, $p = .022$, and question 7 ("The video was interesting."), $F(5, 451) = 3.543$, $p = .004$. All other effects were non-significant.

CMVE shows competitive performance with human edited videos in compilation-style edits. Despite using different and fewer source clips than a professional human editor, CMVE edits generated statistically insignificant quality ratings in a majority of video topics. While we were unable to verify the amount of time required for a human to edit our training sample videos, CMVE performed the most

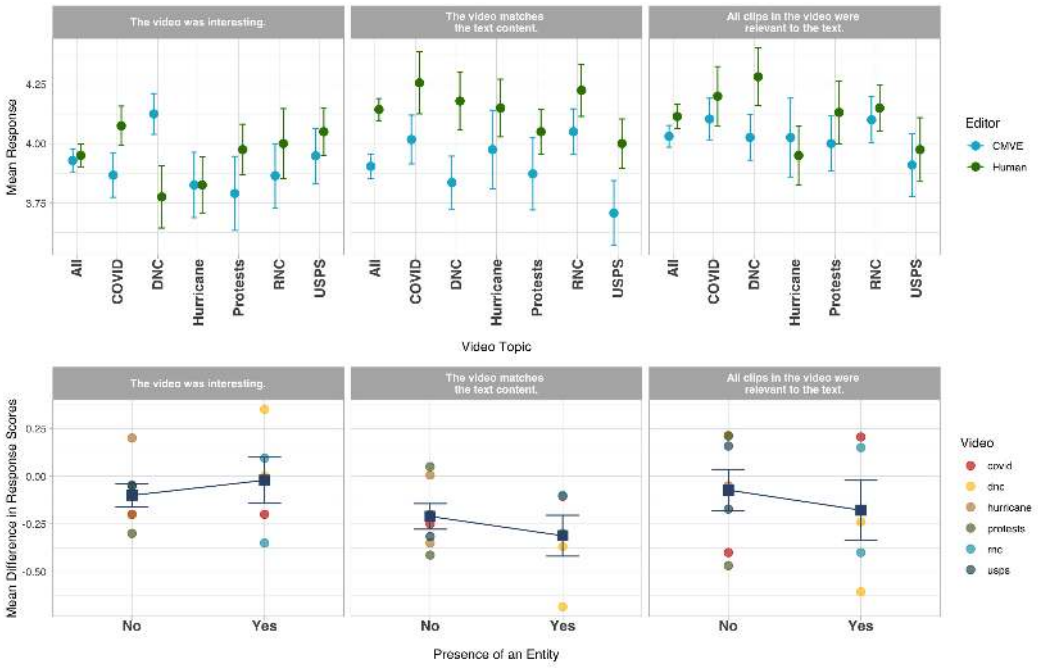

Figure 6. **MTurk Results** - Quality rating results (top) across 6 topics edited with CMVE and compared with Human edits. The mean difference between CMVE and Human (more human preference = greater negative value) vs. the presence of a known entity in the input text query $w^{(\mathcal{I})}$, as recognized by the object recognition model $\mathcal{M}_O$ (bottom). Complete experimental results can be found in the supplementary materials. Error bars show standard error of the mean.

time-intensive steps of ranking (Eq. 3) in 55.5 seconds and stitching (Eq. 17) in 35.5 seconds on average, per video, on a CPU (AWS Graviton Processor with 3008 RAM).

We next consider our hypothesis that entity overlapping information provides a semantically meaningful way to link text queries to known objects in $\mathcal{M}_O$ is shown in the bottom panel of Figure 6. Two-way repeated measure ANOVAs were conducted to evaluate the effect of entity presence and video topic on quality ratings and found no significant main effect for all questions, though the presence of a known entity in the CMVE edit (modeled in Eq. 10) appeared to influence the relative rating in some judgements. While the results did not definitively confirm our hypothesis, we propose future research with larger sample sizes across video domains (e.g., film, news, education), which may be informative about the importance of entities and objects in perceived video coherency.

6. Limitations

While CMVE models object coherency across clips composing a video, a major limitation of our current framework is the inability to model temporal relationships between clips. Furthermore, assessment of CMVE proved challenging due to the creative nature of video editing. Multiple video edits from a given text query can yield the same perceptual quality ratings, proving the relative judgement of edits difficult. We propose future datasets prioritize multiple ground-truth video edits for a given text prompt, which will allow automated editing pipelines to better model the complex nature of creativity.

7. Conclusion

We have proposed an integrated method to automate video editing, termed CMVE, using a text query, videos with descriptions, and a reference video and text query. This method uses perceptual similarity, object similarity and overlap between entities in the reference text query and an object recognition model to compare and rank video clips. We found competitive results when comparing human participant ratings of CMVE edits to human edited videos in a small training sample of news compilation videos, despite using different video sources to tell the same story.

References

[1] Amazon Rekognition – Video and Image - AWS.

[2] Detecting video segments in stored video - Amazon Rekognition.

[3] Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. Video description: A survey of

methods, datasets, and evaluation metrics. *ACM Computing Surveys*, 52(6):1–28, 2019.

[4] AP. AP Newsroom.

[5] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Céspedes, Steve Yuan, Chris Tar, Yun Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder for English. *EMNLP 2018 - Conference on Empirical Methods in Natural Language Processing: System Demonstrations, Proceedings*, pages 169–174, 2018.

[6] Michael G. Christel, Alexander G. Hauptmann, Howard D. Wactlar, and Tobun D. Ng. Collages as dynamic summaries for news video. *Proceedings of the ACM International Multimedia Conference and Exhibition*, pages 561–569, 2002.

[7] Nicholas Davis, Alexander Zook, Brian O'Neill, Brandon Headrick, Mark Riedl, Ashton Grosz, and Michael Nitsche. Creativity support for novice digital filmmaking. *Conference on Human Factors in Computing Systems - Proceedings*, pages 651–660, 2013.

[8] Jyotishman Deka, Om Prakash Tripathi, and Mohammad Latif Khan. Dual Attention Networks for Multimodal Reasoning and Matching. *Journal of Wetlands Ecology*, 5:40–47, 2011.

[9] Yang Feng, Futang Peng, Xu Zhang, Wei Zhu, Shanfeng Zhang, Howard Zhou, Zhen Li, Tom Duerig, Shih-Fu Chang, and Jiebo Luo. Unifying Specialist Image Embedding into Universal Image Embedding. 2020.

[10] Martin Fowler. Simplifying Video Editing Using Metadata. *IEEE Software*, 19(6):13–17, 2002.

[11] Dustin E.R. Freeman, Stephanie Santosa, Fanny Chevalier, Ravin Balakrishnan, and Karan Singh. Laces: Live Authoring through Compositing and Editing of Streaming Video. pages 1207–1216, 2014.

[12] Noa Garcia and Yuta Nakashima. Knowledge-Based Video Question Answering with Unsupervised Scene Descriptions. pages 1–20, 2020.

[13] Øystein Gilje. Multimodal redesign in filmmaking practices: An inquiry of young filmmakers' deployment of semiotic tools in their filmmaking practice. *Written Communication*, 27(4):494–522, 2010.

[14] Abhinav Gupta, Praveen Srinivasan, Jianbo Shi, and Larry S. Davis. Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos. *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2009*, 2009 IEEE:2012–2019, 2009.

[15] Cristian Hofmann, Nina Hollender, and Dieter W. Fellner. Workflow-based architecture for collaborative video annotation. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5621 LNCS:33–42, 2009.

[16] Ines Montani Honnibal, Matthew. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. 2017.

[17] Yuan Jiahong, Mark Liberman, and Christopher Cieri. Towards an integrated understanding of speaking rate in conversation. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2:541–544, 2006.

[18] Shen Jinhong, Seiya Miyazaki, Terumasa Aoki, and Hiroshi Yasuda. Filmmaking production system with rule-based reasoning. *Image and Vision Computing*, pages 366–371, 2003.

[19] J. Kreiman. Perception of sentence and paragraph boundaries in natural conversation. *Journal of Phonetics*, 10(2):163–175, 1982.

[20] Ilya Sutskever Krizhevsky, Alex and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60.6:84–90, 2017.

[21] Cuong Nguyen, Stephen DiVerdi, Aaron Hertzmann, and Feng Liu. Vremiere: In-headset virtual reality video editing. *Conference on Human Factors in Computing Systems - Proceedings*, 2017-May:5428–5438, 2017.

[22] Jeffrey A. Okun and V. E. S. Susan Zwerman. *The VES handbook of visual effects: industry standard VFX practices and procedures.* Routledge, 2020.

[23] Yingwei Pan, Ting Yao, Houqiang Li, and Tao Mei. Video captioning with transferred semantic attributes. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:984–992, 2017.

[24] Zeeshan Rasheed and Mubarak Shah. Detection and representation of scenes in videos. *IEEE Transactions on Multimedia*, 7(6):1097–1105, 2005.

[25] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:8368–8376, 2019.

[26] Cees G.M. Snoek, Bouke Huurnink, Laura Hollink, Maarten De Rijke, Guus Schreiber, and Marcel Worring. Adding semantics to detectors for video retrieval. *IEEE Transactions on Multimedia*, 9(5):975–986, 2007.

[27] spaCy. Annotation Specifications.

[28] Ba Tu Truong, Svetha Venkatesh, and Chitra Dorai. Automatic genre identification for content-based video categorization. *Proceedings - International Conference on Pattern Recognition*, 15(4):230–233, 2000.

[29] Haonan Yu, Jiang Wang, Zhiheng Huang, Yi Yang, and Wei Xu. Video paragraph captioning using hierarchical recurrent neural networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:4584–4593, 2016.

[30] Ping Yu, Chein Shung Hwang, and Nai Wen Kuo. Similarity retrieval of video database based on 3D Z-string. *2nd International Symposium on Electronic Commerce and Security, ISECS 2009*, 2:56–59, 2009.

[31] Alireza Zareian, Svebor Karaman, and Shih-Fu Chang. Weakly Supervised Visual Semantic Parsing. pages 3733–3742, 2020.

[32] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (1):586–595, 2018.