A Graph-Based Framework to Bridge Movies and Synopses

Yu Xiong[1] Qingqiu Huang[1] Lingfeng Guo[2] Hang Zhou[1] Bolei Zhou[1] Dahua Lin[1]
[1]CUHK - SenseTime Joint Lab, The Chinese University of Hong Kong
[2]University of California, Berkeley
{xy017,hq016,bzhou,dhlin}@ie.cuhk.edu.hk zhouhang@link.cuhk.edu.hk lingfeng_guo@berkeley.edu

Abstract

Inspired by the remarkable advances in video analytics, research teams are stepping towards a greater ambition – movie understanding. However, compared to those activity videos in conventional datasets, movies are significantly different. Generally, movies are much longer and consist of much richer temporal structures. More importantly, the interactions among characters play a central role in expressing the underlying story. To facilitate the efforts along this direction, we construct a dataset called Movie Synopses Associations (MSA) *over* 327 *movies, which provides a synopsis for each movie, together with annotated associations between synopsis paragraphs and movie segments. On top of this dataset, we develop a framework to perform matching between movie segments and synopsis paragraphs. This framework integrates different aspects of a movie, including event dynamics and character interactions, and allows them to be matched with parsed paragraphs, based on a graph-based formulation. Our study shows that the proposed framework remarkably improves the matching accuracy over conventional feature-based methods. It also reveals the importance of narrative structures and character interactions in movie understanding. Dataset and code are available at:* `https://ycxioooong.github.io/projects/moviesyn`

1. Introduction

Among various forms of media, movies are often considered as the best to convey stories. While creating a movie, the director can leverage a variety of elements – the scene, the characters, and the narrative structures – to express. From the perspective of computer vision, movies provide a great arena with a number of new challenges, *e.g*. substantially greater length, richer presentation styles, and more complex temporal structures. Recent studies [23, 26, 31, 32, 24, 16] attempted to approach this problem from different angles, only achieving limited progress.

Over the past decade, extensive studies have been devoted to video analytics. A number of video-based tasks, *e.g*. action recognition [34, 4] and event classification [10], have become active research topics. However, methods devised for these tasks are not particularly suitable for movie understanding. Specifically, for such tasks, visual features, which can be a combination of various cues, are often sufficient for obtaining good accuracies. However, movies are essentially different. A movie is created to tell a story, instead of demonstrating a scene or an event of a certain category. To analyze movies effectively, we need new data, new perspectives, and thus new approaches.

Recently, several datasets are constructed on movies, including LSMDC [26] and MovieGraphs [31]. These datasets, however, are limited in that they are small or have a narrow focus on very short clips, *i.e*. those that last for a few seconds. To facilitate the research in movie understanding, we need a new dataset that is large and diverse, and more importantly allows high-level semantics and temporal structures to be extracted and analyzed. In this work, we construct a large dataset called *Movie Synopses Associations (MSA)* over 327 movies. This dataset not only provides a high-quality detailed synopsis for each movie, but also associates individual paragraphs of the synopsis with movie segments via manual annotation. Here, each movie segment can last for several minutes and capture a complete event. These movie segments, combined with the associated synopsis paragraphs, allow one to conduct analysis with a larger scope and at a higher semantic level.

Figure 1 shows a movie segment and the corresponding synopsis paragraph, where we have two important observations: (1) The story is presented with a flow of events, governed by the underlying narrative structures. The sentences in the synopsis often follow a similar order. (2) The characters and their interactions are the key elements of the underlying story. These two key aspects, namely the dynamic flow of events and the interaction among characters, distinguish movies from those videos in conventional tasks.

In this work, we develop a new framework for matching between movie segments and synopsis paragraphs. Rather than encoding them with feature vectors, we choose to use graphs for representation, which provide a flexible way to capture middle-level elements and the relationships among them. Specifically, the framework integrates two key modules: (1) *Event flow module* for aligning the sequence of shots in a movie segment, each showing a particular event, to the sequence of sentences in a synopsis paragraph.

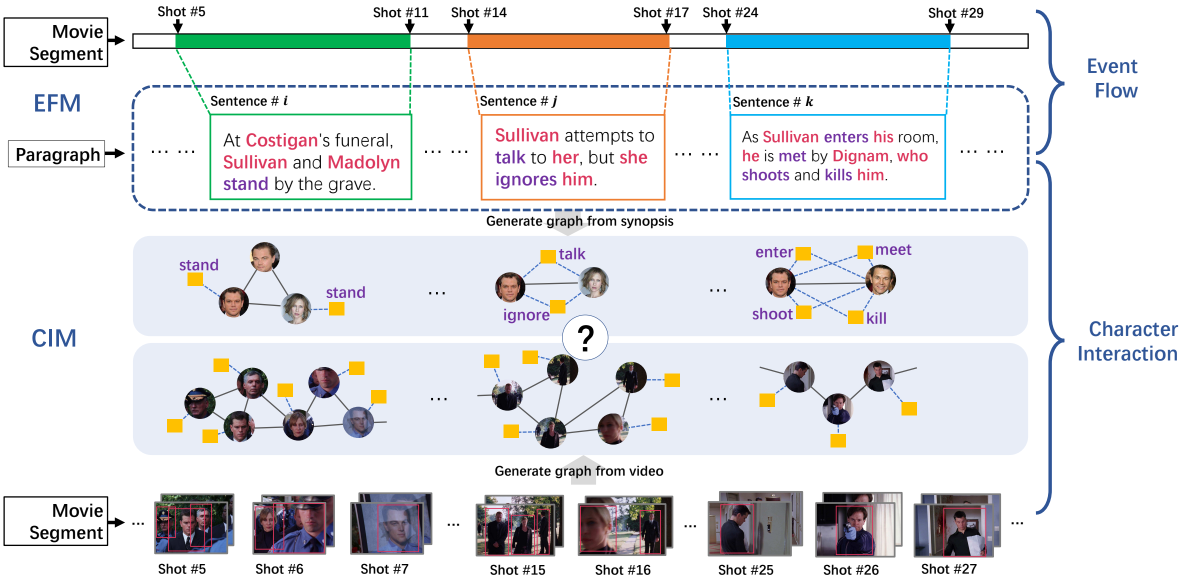


Figure 1. The story in a synopsis paragraph is presented following narrative structures (the upper part), which are modeled into *Event Flow Module*; The lower part shows the character interaction captured in *Character Interaction Module*. The yellow squares denote action.

(2) *Character interaction module* for capturing characters and their behaviors (both actions and interactions) and associating them with the corresponding descriptions. Based on these two modules, the matching can then be done by solving optimization problems formulated based on their respective representations.

It is noteworthy that the use of graphs in movie representation has been explored by previous works [31]. However, our framework is distinguished in several aspects: 1) It takes into account complicated temporal structures and character interactions mined from data. 2) Our method does not require node-to-node annotation when using graphs.

In summary, our contributions lie in three aspects: (1) We construct a large dataset *MSA* on 327 movies, which provides annotated associations between movie segments and synopsis paragraphs. This dataset can effectively support the study on how movie segments are associated with descriptions, which we believe is an important step towards high-level movie understanding. (2) We develop a graph-based framework that takes into account both the flow of events and the interactions among characters. Experiments show that this framework is effective, significantly improving the retrieval accuracies compared to popular methods like visual semantic embedding. (3) We perform a study, which reveals the importance of high-level temporal structures and character interactions in movie understanding. We wish that this study can motivate future works to investigate how these aspects can be better leveraged.

2. Related Work

Datasets for Cross Modal Understanding. In recent years, with the increasing popularity of cross-modal understanding tasks, *e.g.* video retrieval by language, a large number of datasets have been proposed [36, 1, 26, 19, 31, 30, 29, 33]. *ActivityNet Captions* [19] is a dataset with dense captions describing videos from *ActivityNet* [3], which can facilitate tasks such as video retrieval and temporal localization with language queries. *Large Scale Movie Description Challenge (LSMDC)* [26] consists of short clips from movies described by natural language. *MovieQA* [30] is constructed for understanding stories in movies by question answering. Some of the movies are provided plots with aligned movie clips. *MovieGraphs* [31] is established for human-centric situation understanding with graph annotations. But there are three problems for these datasets: (1) most of them obtain dull descriptions from crowd-sourcing platforms, (2) they simply describe short video clips lasting a few seconds, which leads to a huge gap between proposed data and real-world data where the video is much longer and the description is much more complex. (3) some of them are relatively smaller in terms of dataset size. In order to explore the high-level semantics and temporal structures in the data from real-world scenarios, we build a new dataset with long segments cut from movies and diverse descriptions from the synopses in IMDb[1].

Feature-based Methods. To retrieve a video with natural language queries, the main challenge is the gap between two different modals. *Visual Semantic Embedding* (VSE) [9, 7], a widely adopted approach in video retrieval [38, 18, 37, 6, 35], tries to tackle this problem by embedding multi-modal information into a common space. JSF proposed in [37] learns matching kernels based on fea-

[1] https://www.imdb.com

ture sequence fusion. To retrieve video and localize clips, [27] introduces a framework that first perform paragraph level retrieval and then refine the features by sentence level clip localization. Feature-based approaches can not further improve retrieval performance because these methods fail to capture the internal structures of video and language.

Graph-based Methods. Graph-based methods [17, 21, 31], which build semantic graphs from both language and video and then formulate the retrieval task as a graph matching problem [2, 41, 39], is also widely used for cross-modal retrieval. Method in [17] generates scene graph from language queries for image retrieval. A graph matching algorithm is proposed by [21] for semantic search in the domain of autonomous driving. The graph matching problem is formulated as LP optimization with ground-truth alignment in optimization constraints. MovieGraphs proposed in [31] uses graph as semantic representation and integrates graph into potential functions for training. It's noteworthy that node-level annotations are required during training. In this work, we also use graph-based representations for both movies and synopses. However, unlike previous works that depend on the costly node-level annotations, our graph matching only needs ground-truth of paragraph-level alignment, which makes it much more practical.

3. MSA Dataset

This section presents *Movie Synopsis Association (MSA)*, a new dataset constructed upon 327 movies. Particularly, we choose a set of high-quality synopses from IMDb, *i.e*. those with detailed descriptions of individual events, one for each movie. Each synopsis here consists of tens of paragraphs, each describing an event in the movie.

We also provide the associations between movie segments and synopsis paragraphs through manual annotation. These associations constitute a solid basis to support high-level semantic analysis. We collected the associations following the procedure below. (1) We provide the annotators with a complete overview of each movie, including the character list, reviews, *etc*., to ensure they are familiar with the movies. (2) We carry out the annotation procedure in two stages, from coarse to fine. At the first stage, each movie is divided into 64 clips, each lasting for around 2 minutes. For each synopsis paragraph, an annotator is asked to select a segment, *i.e*. a subsequence of N consecutive clips, that cover the corresponding description. At the second stage, annotators adjust the temporal boundaries of the resultant segments to make them better aligned with the paragraphs. This two-stage procedure leads to a collection of paragraph-segment pairs. (3) We dispatch each paragraph to three annotators and only retain those annotations with high consistency among them. Here, the consistency is measured in terms of temporal IoU among the annotations. Finally, we obtained $4,494$ highly consistent paragraph-segment pairs (out of $5,725$ annotations of the original collection).

Table 1. Statistics of the *MSA* dataset.

	Train	Val	Test	Total
# Movies	249	28	50	327
# Segments	3329	341	824	4494
# Shots / seg.	96.4	89.8	76.9	92.3
Duration / seg.	427.4	469.6	332.8	413.3
# Sents. / para.	6.0	6.0	5.5	5.9
# Words. / para.	130.8	132.5	120.5	129.0

Table 2. Comparison between MSA dataset and MovieQA [30].

	#movie	#sent./movie	#words/sent.	dur. (s)
MovieQA	140	35.2	20.3	202.7
MSA	327	81.2	21.8	413.3

Table 1 shows some basic statistics of the dataset. This dataset is challenging: (1) The duration of each movie segment is over 400 seconds on average, far longer than those in existing datasets like LSMDC [26]. (2) The descriptions are rich with over 100 words per paragraph.

Figure 2 compares ActivityNet Caption [19] with the MSA dataset with examples. We can see that the descriptions in MSA are generally much richer and at a higher level, *e.g*. describing characters and events, instead of simple actions. MovieQA also contains description-clip pairs. Table 2 compares MovieQA with our MSA dataset. Note that the plot synopses from MovieQA are obtained from Wikipedia while ours are from IMDb. Compared to synopses from Wikipedia, those from IMDb are written by movie fans and reviewed by others. They are longer and contain more details.

4. Methodology

In this section, we would present our framework for matching between movie segments and synopsis paragraphs. Specifically, given a query paragraph P from a synopsis, we aim at retrieving its associated movie segment Q out of a large pool of candidates. This framework consists of two modules: a *Event Flow Module (EFM)* to exploit the temporal structure of the event flows, and a *Character Interaction Module (CIM)* to leverage character interactions.

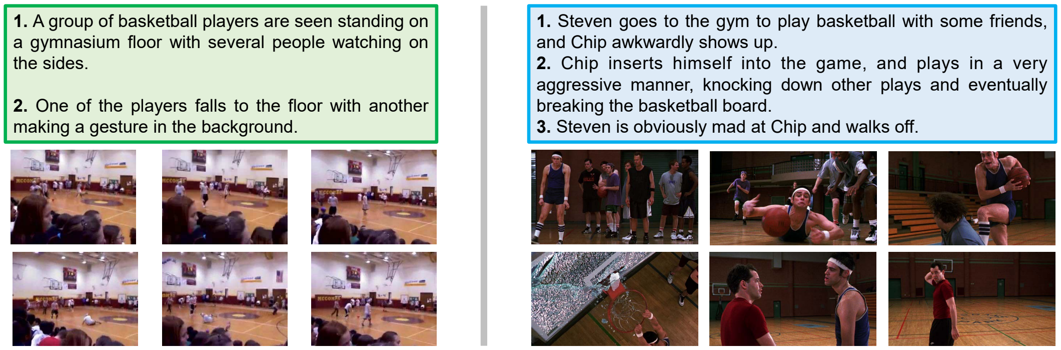
As shown in Figure 1, given a query paragraph P and a candidate movie segment Q, each module yields a similarity score between P and Q, denoted as $\mathcal{S}_{efm}(P,Q)$ and $\mathcal{S}_{cim}(P,Q)$ respectively. Then the overall matching score $\mathcal{S}(P,Q)$ is defined to be their sum as

$$\mathcal{S}(P,Q) = \mathcal{S}_{efm}(P,Q) + \mathcal{S}_{cim}(P,Q), \tag{1}$$

In what follows, Sec. 4.1 and 4.2 present the EFM and CIM modules respectively. Sec. 4.3 introduces the training algorithm, where both modules are jointly optimized.

4.1. Event Flow Module

This module takes into account the temporal structures of event flows. It is motivated by the observation that the sen-

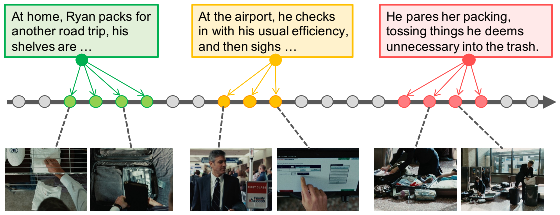


1. A group of basketball players are seen standing on a gymnasium floor with several people watching on the sides.

2. One of the players falls to the floor with another making a gesture in the background.

1. Steven goes to the gym to play basketball with some friends, and Chip awkwardly shows up.
2. Chip inserts himself into the game, and plays in a very aggressive manner, knocking down other plays and eventually breaking the basketball board.
3. Steven is obviously mad at Chip and walks off.

Figure 2. Comparison between examples from ActivityNet Caption (left) and MSA (right). The durations are 12s and 220s respectively.



At home, Ryan packs for another road trip, his shelves are …

At the airport, he checks in with his usual efficiency, and then sighs …

He pares her packing, tossing things he deems unnecessary into the trash.

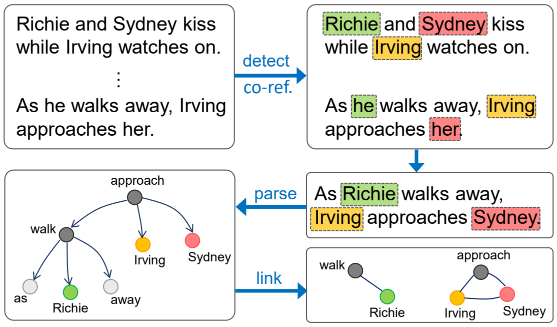
Figure 3. Sentences in a synopsis paragraph often follow a similar order as the situations in event presented in the movie segment. Therefore, they can be aligned temporally.

tences in a synopsis paragraph tend to follow a similar order as that of situation in events (each captured by a sequence of movie shots[2]), as shown in Figure 3. In particular, the alignment between the sentences and the movie shots can be done based on the following principles: (1) Each sentence can match multiple shots while a shot can be assigned to at most one sentence. (2) The sentences and the movie shots follow the same order. The matching should not swap the order, *e.g.* associating a sentence that comes next to a preceding shot.

Formulation. Suppose a paragraph P is composed of a sequence of sentences $\{p_1, \ldots, p_M\}$. We obtain an embedding feature $\boldsymbol{\phi}_i \in \mathbb{R}^D$ for each sentence p_i using fully connected embedding networks. Meanwhile, a movie segment Q consists of a sequence of *shots*, which can be extracted by a shot segmentation tool [28]. We derive a visual feature $\boldsymbol{\psi}_i \in \mathbb{R}^D$ for each shot q_i with fully connected embedding networks. Here we aim at assigning each sentence to a sub-sequence of shots, which can be represented by a binary assignment matrix $\mathbf{Y} \in \{0,1\}^{N \times M}$, where $y_{ij} = \mathbf{Y}(i,j) = 1$ if the i^{th} shot is attached to the j^{th} sentence and 0 otherwise. Given the assignment matrix $\mathbf{Y}$, the total matching score can be expressed as

$$\mathcal{S}_{efm} = \sum_i \sum_j y_{ij} \boldsymbol{\phi}_j^T \boldsymbol{\psi}_i = \operatorname{tr}(\boldsymbol{\Phi}\boldsymbol{\Psi}^T\mathbf{Y}), \qquad (2)$$

where $\boldsymbol{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_M]^T$ and $\boldsymbol{\Psi} = [\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_N]^T$ are



Richie and Sydney kiss while Irving watches on.
⋮
As he walks away, Irving approaches her.

Richie and Sydney kiss while Irving watches on.

As he walks away, Irving approaches her.

As Richie walks away, Irving approaches Sydney.

Figure 4. The procedure of constructing graphs from paragraph. At first, all the character names and pronouns are detected. Then each sentence is parsed to a dependency tree. Based on the tree structure, graphs are generated at rule-based linking stage.

the feature matrices for both domains. Taking the alignment principles described above into account, we can obtain the assignment $\mathbf{Y}$ by solving the following problem:

$$\max_{\mathbf{Y}} \quad \operatorname{tr}(\boldsymbol{\Phi}\boldsymbol{\Psi}^T\mathbf{Y}) \qquad (3)$$

$$\text{s.t.} \quad \mathbf{Y}\mathbf{1} \preceq \mathbf{1}, \qquad (4)$$

$$\mathcal{I}(\mathbf{y}_i) \leq \mathcal{I}(\mathbf{y}_{i+1}), \forall i \leq N-1. \qquad (5)$$

Here, $\mathbf{y}_i$ refers to the i^{th} row of matrix $\mathbf{Y}$, and $\mathcal{I}(\cdot)$ denotes for the index of the first nonzero element in a binary vector. This is a bipartite graph matching problem which can be efficiently solved by dynamic programming.

4.2. Character Interaction Module

As discussed earlier, the interactions among characters play a significant role in movie storytelling. We also observe that the character interactions are often described in synopsis. To incorporate this aspect, we propose the *Character Interaction Module (CIM)* based on graph representations derived from both the synopsis paragraphs and the visual observations in the movie segments.

Specifically, each paragraph and movie segment are represented by graphs $\mathcal{G}_p = (V_p, E_p)$ and $\mathcal{G}_q = (V_q, E_q)$ respectively. The vertex sets V_p and V_q contain both character and action nodes. The edge sets E_p and E_q capture both character-character and character-action relations.

[2] A shot is a series of frames, that runs for an uninterrupted period of time. Observing that frames within a shot are highly redundant, we use shot as the unit instead of frames.

With these two graphs, the similarity between P and Q can be computed by matching between $\mathcal{G}_p$ and $\mathcal{G}_q$. Below, we elaborate on the matching procedure.

Visual Graph from a Movie Segment. Firstly, we generate the character and action nodes: (1) For character nodes, we utilize Faster-RCNN [11] implemented on [5] and pre-trained on [15, 14] to detect person instances in every shot. (2) We attach each person instance with an action node, which comes from a TSN [34] pretrained on *AVA* dataset [12]. Secondly, we produce the edge sets by the following procedures: (1) If a group of people appear in the same or adjacent shots, we introduce an edge between every pair of them. (2) We link each character node to its corresponding action node.

Semantic Graphs from Sentences. For each paragraph, we construct a collections of sub-graphs from each sentence based on dependency trees, as illustrated in Figure 4.

The construction process consists of four major steps: **(1) Name detection:** We detect all the named entities (*e.g.*, *Jack*) using StanfordNer [8]. Then we resort to CorefAnnotator [25] to link pronouns with named entities and substitute all pronouns with their corresponding names. **(2) Character association:** With the help of IMDb, we can retrieve a portrait for each named character and thus obtain facial and body features using ResNet [13] pre-trained on *PIPA* [40]. This allows character nodes to be matched to the person instances detected in the movie. **(3) Sentence parsing:** We use GoogleNLP API[3] to obtain the dependency tree of a sentence. Each node in the tree is labeled with a part-of-speech tagging. **(4) Edge linking:** Based on the dependency tree, we link each character name to its parent verb. Meanwhile, if a group of character names share the same verb, we introduce an edge between every pair of them. Note that we only consider the verbs that stand for action. We first select 1000 verbs with the highest frequency from the synopses corpus, and then retain those corresponding to visually observable actions, *e.g.* "run". This results in a set of 353 verbs.

It is worth noting that we generate a collection of sub-graphs from paragraph instead of a connected graph. For convenience, we consider the collection of sub-graphs as a graph with notation $\mathcal{G}_p$ although it can be further decomposed into multiple disjoint sub-graphs. This is also what we do in our implementation.

Matching Paragraph with Movie Segment. For graph $\mathcal{G}_p$, let V_p be its vertex set with $|V_p| = m = m_c + m_a$, where m_c is the number of character nodes and m_a is that of action nodes. Similarly, we have $\mathcal{G}_q$ with $|V_q| = n = n_c + n_a$.

The target of graph matching is to establish a node-to-node assignment for the two input graphs while taking the the pair-wise constraints, namely the edges, into account.

We define a binary vector $\mathbf{u} \in \{0,1\}^{nm\times 1}$ as the indicator, where $u_{ia} = 1$ if $i \in V_q$ is assigned to $a \in V_p$. To measure the similarity of nodes and edges from different graphs, we establish the similarity matrix $\mathbf{K} \in \mathbb{R}^{nm\times nm}$, where the diagonal elements represent node similarities whereas the off-diagonal entries denote edge similarities. Particularly, $\kappa_{ia;ia} = \mathbf{K}(ia, ia)$ measures the similarity between i^{th} node in V_q and a^{th} node in V_p. $\kappa_{ia;jb}$ measures the similarity between two edges $(i,j) \in E_q$ and $(a,b) \in E_p$. The nodes are represented as output features from networks. And the edge is represented by the concatenation of its nodes' features. The similarities in $\mathbf{K}$ is computed by dot product between feature vectors.

Given the indicator $\mathbf{u}$ and the similarity matrix $\mathbf{K}$, the similarity of two graphs can be derived as

$$\mathcal{S}_{cim}(P,Q) = \sum_{i,a} u_{ia}\kappa_{ia;ia} + \sum_{\substack{i,j\\ i\neq j}} \sum_{\substack{a,b\\ a\neq b}} u_{ia}u_{jb}\kappa_{ia;jb}, \quad (6)$$

where the first term models the similarity score between matched notes $i \in V_q$ and $a \in V_p$. The second term gives the bonus from matched edges between (i,j) and (a,b).

Based on the properties of nodes, certain constraints are enforced on $\mathbf{u}$: (1) The matching should be a one-to-one mapping. For example, one node in a vertex set can only be matched to at most one node in the other set. (2) Nodes of different types cannot be matched together. For example, a character node can not be assigned to an action node.

The objective function, together with the constraints, can be simply expressed in the following form:

$$\max_{\mathbf{u}} \quad \mathbf{u}^T\mathbf{K}\mathbf{u}, \quad (7)$$

$$\text{s.t.} \quad \textstyle\sum_i u_{ia} \leq 1 \quad \forall a, \quad (8)$$

$$\textstyle\sum_a u_{ia} \leq 1 \quad \forall i, \quad (9)$$

$$\textstyle\sum_{i\in V_q^c} u_{ia} = 0 \quad \forall a \in V_p^a, \quad (10)$$

$$\textstyle\sum_{i\in V_q^a} u_{ia} = 0 \quad \forall a \in V_p^c. \quad (11)$$

Here V_q^a denotes the vertex set containing only action nodes in video with $|V_q^a| = n_a$ and V_q^c for vertex only containing cast nodes in video. The same for V_p^a and V_p^c.

Graph Pruning The problem itself is known as an NP-hard *Quadratic Assignment Problem (QAP)*. Solving it could be time consuming especially when the graph is large, which is normally the case for our video graph. To ease the problem, we propose a graph pruning strategy to reduce the graph size to an appropriate one that it can be solved in an affordable time. The strategy is described as follows:

Seed Node Generation. We first select the most important nodes as seed nodes. They are selected by the following two criteria: (a) The problem can be approximately solved by *Kuhn–Munkres (KM)* algorithm [20] in polynomial time. The matched nodes can be selected as seed nodes. (b) The

[3] https://cloud.google.com/natural-language/

k most similar nodes with each node from the query graph will be chosen as seed nodes.

Selection Propagation. Given the seed nodes, we extend the node selection by considering the nodes within J^{th} degree connection of a seed node. We denote the seed nodes by another indicator vector $\mathbf{v} \in \{0,1\}^{n\times 1}$, the adjacency matrix as $\mathcal{A}$ of graph $\mathcal{G}_q$, the nodes we select can be expressed as $\mathbf{v} \leftarrow \mathcal{A}^J\mathbf{v}$. The pruned graph is obtained by cropping the whole graph using selected nodes.

4.3. Joint Optimization.

The quality of the node features would highly influence the result of matching. It is necessary for us to finetune the parameters of the models in EFM and CIM for a better representations. Since we do not have the ground truth alignment of $\mathbf{Y}$ in EFM or $\mathbf{u}$ in CIM, we can not directly update the model parameters in a supervised manner. Hence, we adopt an EM-like procedure to finetune the feature representations and optimize matching objectives. The overall loss of the whole framework is given below:

$$\mathcal{L} = \mathcal{L}(\mathbf{Y}, \boldsymbol{\theta}_{efm}, \mathbf{u}, \boldsymbol{\theta}_{cim}) \tag{12}$$

where $\boldsymbol{\theta}_{efm}$ and $\boldsymbol{\theta}_{cim}$ denote model parameters for embedding networks in EFM and CIM respectively.

E-Step. Using current model parameter values $\boldsymbol{\theta}^*_{efm}$ and $\boldsymbol{\theta}^*_{cim}$, we solve Eq.3 by dynamic programming mentioned in Sec.4.1 and we obtain a sub-optimal value in Eq.7 by applying KM algorithm. Here in our implementation, the time complexity of the KM algorithm is $\mathcal{O}(\tau^3)$ where $min(n,m) \leq \tau \leq max(n,m)$.

M-Step. We update the model parameters in M-step with optimal solutions $\mathbf{Y}^*$ and $\mathbf{u}^*$ obtained in E-step. Particularly, given $\mathbf{Y}^*$ and $\mathbf{u}^*$, we update model parameters by

$$\begin{aligned}\boldsymbol{\theta}^*_{efm}, \boldsymbol{\theta}^*_{cim} &= \underset{\boldsymbol{\theta}_{efm},\boldsymbol{\theta}_{cim}}{\text{argmin}} \ \mathcal{L}(\mathbf{Y}^*, \boldsymbol{\theta}_{efm}, \mathbf{u}^*, \boldsymbol{\theta}_{cim}) \\ &= \underset{\boldsymbol{\theta}_{efm},\boldsymbol{\theta}_{cim}}{\text{argmin}} \ \mathcal{L}(S^*; \boldsymbol{\theta}_{efm}, \boldsymbol{\theta}_{cim})\end{aligned} \tag{13}$$

where $\mathcal{L}(S;\boldsymbol{\theta})$ is the pair-wise ranking loss with margin α shown below:

$$\begin{aligned}\mathcal{L}(S;\boldsymbol{\theta}) = \sum_i \sum_{j\neq i} max(0, S(Q_j, P_i) - S(Q_i, P_i) + \alpha) \\ + \sum_i \sum_{j\neq i} max(0, S(Q_i, P_j) - S(Q_i, P_i) + \alpha)\end{aligned} \tag{14}$$

5. Experiments

We conduct experiments of movie-synopsis retrieval on MSA dataset. Specifically, search a movie segment from candidate pool given a synopsis paragraph as query.

5.1. Experiment Setup

Dataset. The MSA dataset is randomly split into *train, val, test* subsets with $3329, 341, 824$ samples respectively. Note that there are no overlap movies among subsets. The statistic of the subsets is shown in Table 1.

There are two settings to measure the performance, namely, **cross-movie** and **within-movie**. The cross-movie setting considers the whole test set as the candidate pool for each query whereas the within-movie setting only takes the segments from the same queried movie to be the candidates.

Evaluation Metrics. To evaluate the performance, we adopt the commonly used metrics: (1) **Recall@K**: the fraction of GT videos that have been ranked in top K; (2) **MedR**: the median rank of GT videos. (3) **Avg. MedR**: Average MedR, this is only for within-movie setting.

Implementation Details. In EFM, Word2Vec [22] embedding is used as sentence representation. The Word2Vec model is finetuned on MSA corpus, *i.e.*, synopses and subtitles. The shot feature consists of two parts: 1) visual features extracted from $pool5$ layer of ResNet-101 [13]. 2) its subtitle's Word2Vec embedding. In CIM, we adopt ResNet-50 pre-trained on *PIPA* [40] to extract the face and body feature for a detected person instance or a cast portrait. The action features in videos come from TSN [34] pre-trained on AVA [12] and action verbs are represented by Word2Vec embeddings. We train all the embedding networks using SGD with learning rate 0.001. The batch size is set to 16 and the margin α in pair-wise ranking loss is set to 0.2.

5.2. Overall Results

We adopt VSE as the base models and previous method JSF [37] is also used for comparison. Also for comparison, we gradually add three kinds of features, namely, *appearance, cast* and *action* as nodes to baseline method. Particularly, appearance node denotes the sentence embeddings or shot features. For VSE, the features of movie shots and sentences are further transformed with two-layer MLPs. We then obtain the features of segments and paragraphs by taking the average of the shot and sentence features. During matching, the segment/paragraph similarities are computed with cosine similarity. We use the same loss as shown in Eq. 14. Matching scores from different nodes are fused by weighted sum. The weights are obtained by observing the performance of single node on val set. Here, for cross-movie setting, weights are simply set as $0.3, 1.0$ and 0.1 for appearance, cast and action respectively. For within-movie setting, weights are $0.3, 0.3$ and 0.1. Table 3 shows the overall results of video retrieval on MSA.

Analysis on Overall Results. From the results shown in Table 3, by comparing different methods, we observe that:

(1) Both VSE and JSF outperform random guess by a large margin. The performance of JSF does not exceed

Table 3. The overall performance of video retrieval on MSA dataset under both cross-movie and within-movie settings. Here, *appr.* refers to appearance node, *cast* stands for character node and *action* denotes action node.

			Cross-movie				Within-movie			
	Method	Nodes	R@1	R@5	R@10	MedR	R@1	R@5	R@7	Avg. MedR
1	Random	-	0.12	0.61	1.21	412.5	6.07	28.88	38.35	8.74
2	JSF	appr.	3.52	12.62	20.02	55	19.42	56.07	66.51	3.86
3	VSE	appr.	4.49	15.41	24.51	39.5	21.36	60.07	69.42	3.62
4	VSE	appr.+action	5.34	15.78	24.64	42.5	21.85	61.41	69.66	3.47
5	VSE	appr.+action+cast	19.05	48.67	60.92	6	26.70	65.90	72.94	3.03
6	Ours(EFM)	appr.	6.80	20.15	28.40	36	27.67	63.59	71.97	2.92
7	Ours(EFM)	appr.+action+cast	21.12	48.67	61.04	6	30.58	66.14	73.42	2.70
8	Ours(EFM+CIM)	appr.+action+cast	**24.15**	**53.28**	**66.75**	**4.5**	**31.92**	**67.96**	**74.76**	**2.50**

Table 4. Influence of different choices of N for updating scores in CIM. The first row is the result before updating.

	R@1	R@5	R@10	MedR
previous stage	21.12	48.67	61.04	6
$N = 15$	**24.15**	**53.28**	**66.75**	**4.5**
$N = 40$	23.91	51.94	63.71	5
$N = 60$	23.42	51.46	63.11	5
$N = 80$	23.42	51.46	62.86	5

Table 5. Comparison between the performance of using only visual feature and that of using both visual and subtitle features as shot representation. The input node is *appr.*

	R@1	R@5	R@10	MedR
visual only	4.25	13.84	19.66	56
visual + subtt.	**4.49**	**15.41**	**24.51**	**39.5**

Table 6. Comparison of different graph pruning parameters.

	R@1	R@5	R@10	MedR
$J = 1$	23.30	53.03	66.14	5
$J = 2$	**24.15**	**53.28**	**66.75**	**4.5**
$J = 3$	24.03	53.16	66.63	5

that of VSE because the learned kernels in JSF fail to capture the matching pattern between paragraphs and long videos, when the concepts in paragraphs are complicated and lengths of videos vary a lot.

(2) Our method with EFM and CIM outperforms the conventional methods that only fuse features under both cross-movie and within-movie settings. Particularly, Recall@1 under cross-movie setting is raised from 19.05% to 24.15% (5.10% absolute and 27% relative improvement) and each recall under within-movie setting improves over 1.5%.

Analysis on EFM and CIM. Also shown in Table 3, the results of rows 3,6 demonstrate that the proposed EFM improves the performance on most of the metrics. We can see from the table that EFM works better especially under within-movie setting (6.31% increment on Recall@1). It is because that encoded story and narrative structure in EFM is the key to distinguish segments from the same movie.

Meanwhile, results from rows 7-8 prove the effectiveness of using character interaction graph, especially under cross-movie setting. The CIM does not bring consistent performance gain under within-movie setting compared to EFM. The reason is that segments from the same movie share a group of characters and their interactions are also similar. This is also illustrated in the right part of rows 4-5.

5.3. Ablation Studies

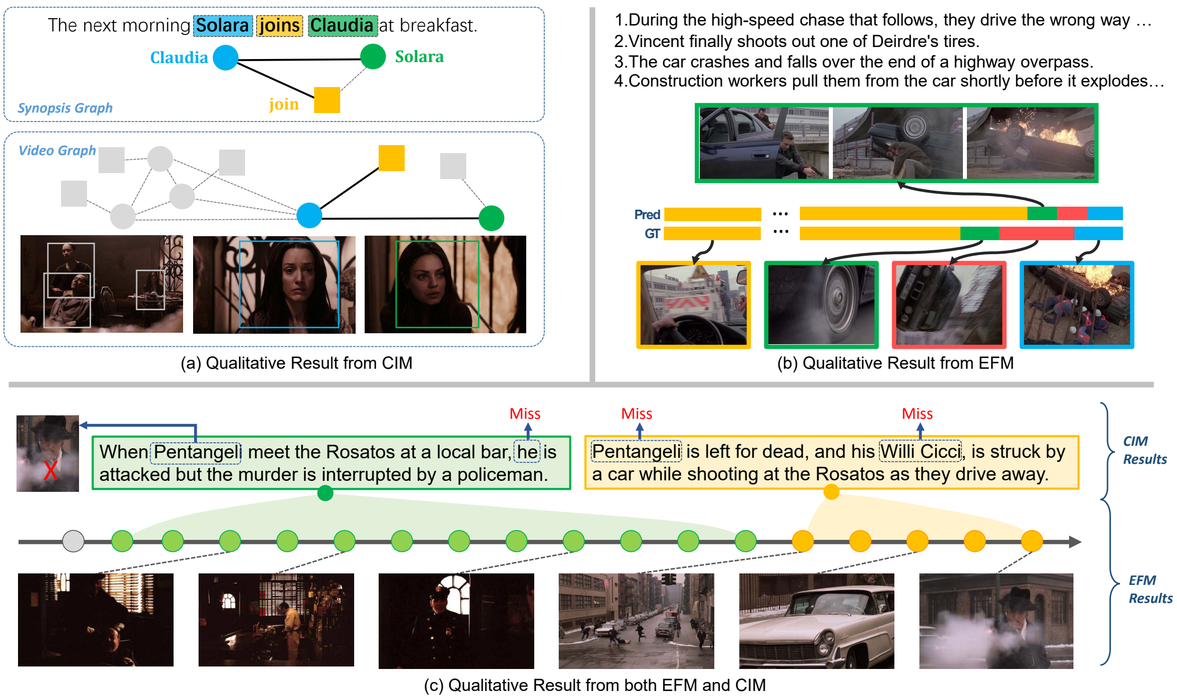
We present ablation studies on different hyper parameters. Unless stated, experiments are conducted under cross-movie setting.

Choices of N in CIM. As mentioned before, at inference stage, we need to obtain score in CIM by solving the optimization problem in Eq.7. It takes 2 seconds to solve one matching on average. Under the cross-movie setting, we need to solve these problems for 824^2 times (the number of test samples is 824), which sums up to more than a week. To save time, we only update the score of candidates that rank top N in previous stage, *e.g.*, VSE with score fusion.

Table 4 shows the influence on different choices of N. Note that we take the score in the first row to filter out a candidate list for updating. We see that from $N = 15$ to $N = 40$, the performance drops while remains steady when N increases from 40 to 80. All the results still outperform the baseline in the first row. The performance drop comes from the increasing outliers when N increases. Therefore, decrease N can not only improve inference efficiency but also decrease the number of distractors in candidate pool.

Influence of using subtitle feature. Recall that we use both the visual and subtitle feature as the representation of a shot by observing that sometimes the narrators tend to summarize important dialogues in synopses. We conduct ablation study on the effectiveness of subtitle feature shown in Table 5. The experiments are based on appearance nodes only. The results show that subtitle are complementary to visual information.

Graph Pruning Parameters. To raise inference efficiency, we perform graph pruning in CIM. We set $k = 2$ to select seed and $J = 2$ to spread selection (recall Sec. 4.2). As k and J are complementary for controlling the size of pruned graph, we only conduct studies on different values



The next morning Solara joins Claudia at breakfast.

(a) Qualitative Result from CIM

1.During the high-speed chase that follows, they drive the wrong way …
2.Vincent finally shoots out one of Deirdre's tires.
3.The car crashes and falls over the end of a highway overpass.
4.Construction workers pull them from the car shortly before it explodes…

(b) Qualitative Result from EFM

When Pentangeli meet the Rosatos at a local bar, he is attacked but the murder is interrupted by a policeman.

Pentangeli is left for dead, and his Willi Cicci, is struck by a car while shooting at the Rosatos as they drive away.

(c) Qualitative Result from both EFM and CIM

Figure 5. Qualitative results of EFM and CIM modules. (a) shows a success case of CIM; (b) presents a failure case of EFM; (c) shows an example that EFM succeeds but CIM fails.

of J. The results are shown in Table 6. It demonstrates that $J = 2$ is enough for pruning a graph and increase J may introduce more noise.

5.4. Qualitative Results

We present qualitative results on both EFM and CIM modules to further explore their effectiveness.

Figure 5 (a) shows a positive result that the characters and actions in the sentence are accurately matched. The right matching is obtained with the help of character-character and character-action relations.

Figure 5 (c) shows a case that EFM successfully assigns each sentence to the corresponding shots while CIM fails to assign the characters. In particular, "Pentangeli" is assigned to a wrong person instance while the other three names match nothing. The reason is that the person instances from movie segment are in poor quality due to dim light, occlusion or large motion expect the one appearing at the end of the segment.

Figure 5 (b) shows a failure case of EFM where the second sentence is completely miss-aligned. As shown in the upper part of the figure, this is possible because the shots belong to the third sentence contain some content of "shoot" and "tire" which mislead the model. We also observe that this case is challenging because the shots look similar to each other due to no transition of scene.

From the above observations and analysis on more such cases, we come to the following empirical conclusions: (1) Edge constraints are important for alignments. (2) The quality of nodes matters. If nodes are in poor quality, the edge constraints will take no effect. (3) Discriminative shot appearance, together with our proposed EFM, is helpful for temporal alignment.

6. Conclusion

In this paper, we propose a new framework for matching between movie segments and synopsis paragraphs. The proposed framework integrates a *Event Flow Module* to capture the *narrative structures* of movies and a *Character Interaction Module* to model character interactions using graph-based formulation. To facilitate research for movie-synopsis matching, we construct a dataset called Movie Synopses Associations (*MSA*). Experimental results show the effectiveness of the proposed modules. Our framework outperforms conventional feature-based methods and improves the matching accuracy consistently on all metrics. Both quantitative and qualitative studies demonstrate that our method can capture rich temporal structures and diverse interactions among characters.

7. Acknowledgment

This work is partially supported by the Collaborative Research grant from SenseTime Group (CUHK Agreement No. TS1610626 & No. TS1712093), and the General Research Fund (GRF) of Hong Kong (No. 14236516 & No. 14203518).

References

[1] Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. Localizing moments in video with natural language. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5803–5812, 2017.

[2] Alexander C Berg, Tamara L Berg, and Jitendra Malik. *Shape matching and object recognition using low distortion correspondences*. IEEE, 2005.

[3] Fabian Caba Heilbron, Victor Escorcia, Bernard Ghanem, and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[6] Jianfeng Dong, Xirong Li, Chaoxi Xu, Shouling Ji, and Xun Wang. Dual dense encoding for zero-example video retrieval. *arXiv preprint arXiv:1809.06181*, 2018.

[7] Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612*, 2017.

[8] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[9] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[10] Chuang Gan, Naiyan Wang, Yi Yang, Dit-Yan Yeung, and Alex G Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2568–2577, 2015.

[11] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[12] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[14] Qingqiu Huang, Wentao Liu, and Dahua Lin. Person search in videos with one portrait through visual and temporal links. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 425–441, 2018.

[15] Qingqiu Huang, Yu Xiong, and Dahua Lin. Unifying identification and context learning for person recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[16] Qingqiu Huang, Yuanjun Xiong, Yu Xiong, Yuqi Zhang, and Dahua Lin. From trailers to storylines: An efficient way to learn from movies. *arXiv preprint arXiv:1806.05341*, 2018.

[17] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David Shamma, Michael Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3668–3678, 2015.

[18] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Yiannis Kompatsiaris. Near-duplicate video retrieval with deep metric learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 347–356, 2017.

[19] Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. Dense-captioning events in videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 706–715, 2017.

[20] Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[21] Dahua Lin, Sanja Fidler, Chen Kong, and Raquel Urtasun. Visual semantic search: Retrieving videos via complex textual queries. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2657–2664, 2014.

[22] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[23] Seil Na, Sangho Lee, Jisung Kim, and Gunhee Kim. A read-write memory network for movie story understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 677–685, 2017.

[24] A. Nagrani and A. Zisserman. From benedict cumberbatch to sherlock holmes: Character identification in tv series without a script. In *BMVC*, 2017.

[25] Marta Recasens, Marie-Catherine de Marneffe, and Christopher Potts. The life and death of discourse entities: Identifying singleton mentions. In *North American Association for Computational Linguistics (NAACL)*, 2013.

[26] Anna Rohrbach, Marcus Rohrbach, Niket Tandon, and Bernt Schiele. A dataset for movie description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3202–3212, 2015.

[27] Dian Shao, Yu Xiong, Yue Zhao, Qingqiu Huang, Yu Qiao, and Dahua Lin. Find and focus: Retrieve and localize video events with natural language queries. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[28] Panagiotis Sidiropoulos, Vasileios Mezaris, Ioannis Kompatsiaris, Hugo Meinedo, Miguel Bugalho, and Isabel Trancoso.

Temporal video segmentation to scenes using high-level audiovisual features. *IEEE Transactions on Circuits and Systems for Video Technology*, 21(8):1163–1177, 2011.

[29] Makarand Tapaswi, Martin Bauml, and Rainer Stiefelhagen. Book2movie: Aligning video scenes with book chapters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1827–1835, 2015.

[30] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4631–4640, 2016.

[31] Paul Vicol, Makarand Tapaswi, Lluis Castrejon, and Sanja Fidler. Moviegraphs: Towards understanding human-centric situations from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8581–8590, 2018.

[32] Anran Wang, Anh Tuan Luu, Chuan-Sheng Foo, Hongyuan Zhu, Yi Tay, and Vijay Chandrasekhar. Holistic multimodal memory network for movie question answering. *arXiv preprint arXiv:1811.04595*, 2018.

[33] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5005–5013, 2016.

[34] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.

[35] Huijuan Xu, Kun He, Bryan A. Plummer, Leonid Sigal, Stan Sclaroff, and Kate Saenko. Multilevel language and vision integration for text-to-clip retrieval. In *AAAI*, 2019.

[36] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5288–5296, 2016.

[37] Youngjae Yu, Jongseok Kim, and Gunhee Kim. A joint sequence fusion model for video question answering and retrieval. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[38] Youngjae Yu, Hyungjin Ko, Jongwook Choi, and Gunhee Kim. End-to-end concept word detection for video captioning, retrieval, and question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3165–3173, 2017.

[39] Andrei Zanfir and Cristian Sminchisescu. Deep learning of graph matching. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[40] Ning Zhang, Manohar Paluri, Yaniv Taigman, Rob Fergus, and Lubomir Bourdev. Beyond frontal faces: Improving person recognition using multiple cues. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4804–4813, 2015.

[41] Feng Zhou and Fernando De la Torre. Factorized graph matching. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 127–134. IEEE, 2012.