MovieGraphs: Towards Understanding Human-Centric Situations from Videos

Paul Vicol[1,2] Makarand Tapaswi[1,2] Lluís Castrejón[3] Sanja Fidler[1,2]
[1]University of Toronto [2]Vector Institute [3]Montreal Institute for Learning Algorithms
{pvicol, makarand, fidler}@cs.toronto.edu, lluis.enric.castrejon.subira@umontreal.ca
http://moviegraphs.cs.toronto.edu

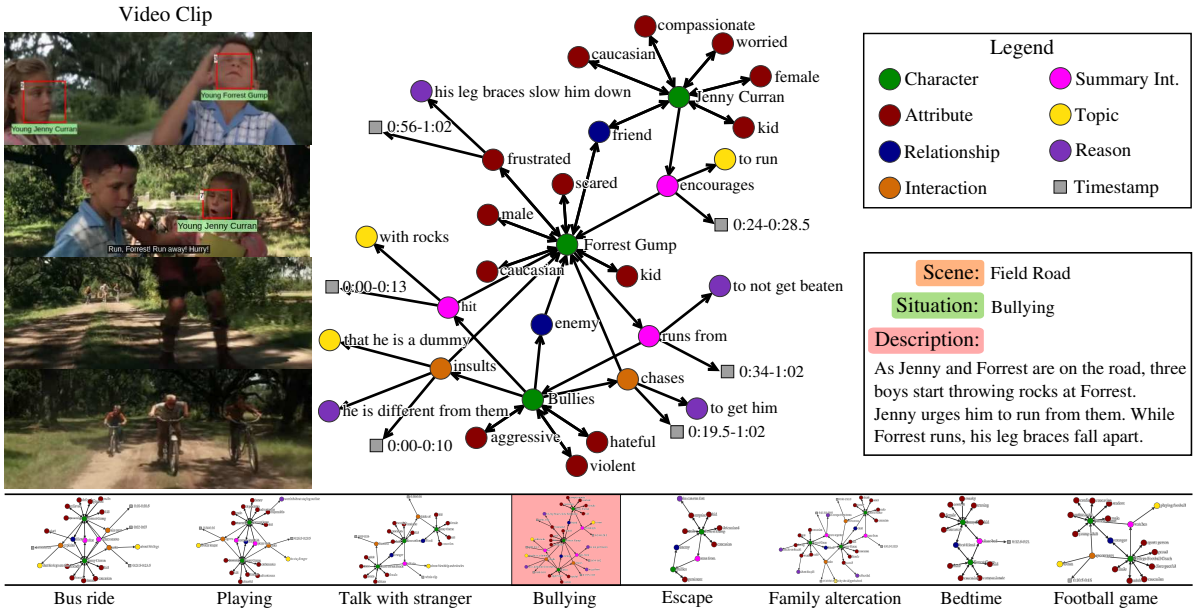

Figure 1: An example from the *MovieGraphs* dataset. Each of the 7637 video clips is annotated with: 1) a *graph* that captures the characters in the scene and their attributes, interactions (with topics and reasons), relationships, and time stamps; 2) a *situation label* that captures the overarching theme of the interactions; 3) a *scene label* showing where the action takes place; and 4) a natural language *description* of the clip. The graphs at the bottom show situations that occur before and after the one depicted in the main panel.

Abstract

There is growing interest in artificial intelligence to build socially intelligent robots. This requires machines to have the ability to "read" people's emotions, motivations, and other factors that affect behavior. Towards this goal, we introduce a novel dataset called MovieGraphs which provides detailed, graph-based annotations of social situations depicted in movie clips. Each graph consists of several types of nodes, to capture who is present in the clip, their emotional and physical attributes, their relationships (i.e., parent/child), and the interactions between them. Most interactions are associated with topics *that provide additional details, and* reasons *that give motivations for actions. In addition, most interactions and many attributes are* grounded *in the video with time stamps. We provide a thorough analysis of our dataset, showing interesting common-sense correlations between different social aspects of scenes, as well as across scenes over time. We propose a method for querying videos and text with graphs, and show that: 1) our graphs contain rich and sufficient information to summarize and localize each scene; and 2) subgraphs allow us to describe situations at an abstract level and retrieve multiple semantically relevant situations. We also propose methods for interaction understanding via ordering, and reason understanding. MovieGraphs is the first benchmark to focus on inferred properties of human-centric situations, and opens up an exciting avenue towards socially-intelligent AI agents.*

1. Introduction

An important part of effective interaction is behaving appropriately in a given situation. People typically know how to talk to their boss, react to a worried parent or a naughty child, or cheer up a friend. This requires proper reading of people's emotions, understanding their mood,

motivations, and other factors that affect behavior. Furthermore, it requires understanding social and cultural norms, and being aware of the implications of one's actions. The increasing interest in social chat bots and personal assistants [1, 4, 18, 22, 27, 42] points to the importance of teaching artificial agents to understand the subtleties of human social interactions.

Towards this goal, we collect a novel dataset called *MovieGraphs* (Fig. 1) containing movie clips that depict human-centric situations. Movies are a rich source of information about behavior, because like people in the real world, movie characters face a variety of situations: they deal with colleagues at work, with family at home, with friends, and with enemies. Past situations lead to new situations, relationships change over time, and we get to see the same character experience emotional ups and downs just as real people do. The behavior of characters depends on their interpersonal relationships (*e.g.* family or friends), as well as on the social context, which includes the scene (*e.g.* bar) and situation (*e.g.* date). We use *graphs* to describe this behavior because graphs are more structured than natural language, and allow us to easily ground information in videos.

The *MovieGraphs* dataset consists of 7637 movie clips annotated with graphs that represent who is in each clip, the interactions between characters, their relationships, and various visible and inferred properties such as the reasons behind certain interactions. Each clip is also annotated with a situation label, a scene label (where the situation takes place), and a natural language description. Furthermore, our graphs are visually and temporally grounded: characters in the graph are associated with face tracks in the clip, and most interactions are associated with the time intervals in which they occur.

We provide a detailed analysis of our dataset, showing interesting common-sense correlations between different social aspects of situations. We propose methods for graph-based video retrieval, interaction understanding via ordering, and understanding motivations via reason prediction. We show that graphs contain sufficient information to localize a video clip in a dataset of movies, and that querying via subgraphs allows us to retrieve semantically meaningful clips. Our dataset and code will be released (http://moviegraphs.cs.toronto.edu), to inspire future work in this exciting domain.

The rest of this paper is structured as follows: in Sec. 2, we discuss related work; Sec. 3 describes our dataset; Sec. 4 introduces the models we use for video retrieval, interaction ordering, and reason prediction; Sec. 5 presents the results of our experiments; and we conclude in Sec. 6.

2. Related Work

Video Understanding. There is increasing effort in developing video understanding techniques that go beyond classifying actions in short video snippets [19, 26], towards parsing more complex videos [5, 35, 36]. A large body of work focuses on identifying characters in movies or TV series [6, 10, 33, 38] and estimating their poses [9]. Steps towards understanding social aspects of scenes have included classifying four visual types of interactions [31], and predicting whether people are looking at each other [25]. [8, 29] find communities of characters in movies and analyze their social networks. In [11], the authors predict coarse social interaction groups (*e.g.* monologue or dialog) in ego-centric videos collected at theme parks. In the domain of affective computing, the literature covers user studies of social behavior [14]. However, we are not aware of any prior work that analyzes and models human-centric situations at the level of detail and temporal scale that we present here. Additionally, our annotations are richer than in Hollywood2 [2] (action labels vs interaction graphs), and more detailed than Large Scale Movie Description Challenge (LSMDC) [3] (single sentence vs short descriptions).

Video Q&A. Other ways to demonstrate video understanding include describing short movie clips [34, 37, 41, 47] and answering questions about them [13, 16, 28, 40]. However, these models typically form internal representations of actions, interactions, and emotions, and this implicit knowledge is not easy to query. We believe that graphs may lead to more interpretable representations.

Graphs as Semantic Representations. Recently, there has been increasing interest in using graphs as structured representations of semantics. Johnson *et al.* [15] introduce *scene graphs* to encode the relationships between objects in a scene and their attributes, and show that such graphs improve image retrieval compared to unstructured text. Recent work aims to generate such scene graphs from images [43].

While retrieval methods using structured prediction exist, ours is the first to use video. Thus the potentials in our model are very different, as we deal with a different problem: analyzing people. Our graphs capture human behavior (*e.g. encourages*) that is inferred from facial expressions, actions, and dialog. In contrast, [15] deals with spatial relationships between objects in images (*e.g. in front of*).

Semantic Role Labeling. [23, 44, 45] deal with recognizing situations in images. This task involves predicting the dominant action (verb) as well as the semantic frame, *i.e.* a set of action-specific roles. However, these works focus on static images with single actions, while we focus on movie clips (videos and dialogs) and tackle different tasks.

3. The MovieGraphs Dataset

We construct a dataset to facilitate machine understanding of real-world social situations and human behaviors. We annotated 51 movies; each movie is first split into scenes

	TRAIN	VAL	TEST	TOTAL
# Movies	34	7	10	51
# Video Clips	5050	1060	1527	7637
Desc #Words	35.53	34.47	34.14	35.11
Desc #Sents	2.73	2.45	2.91	2.73
Characters	3.01	2.97	2.9	2.98
Interactions	3.18	2.48	3.11	3.07
Summary Int.	2	2.06	2.05	2.02
Relationships	3.12	2.77	3.52	3.15
Attributes	13.59	14.53	13.79	13.76
Topics	2.64	2.7	2.68	2.65
Reasons	1.66	1.53	2.17	1.74
Timestamps	4.23	4.34	4.68	4.34
Avg. Duration	43.96	43.90	45.61	44.28

Table 1: Statistics of the MovieGraphs dataset across train, validation, and test splits. We show the number of movies and clips; their average duration (sec); the number of words/sentences in descriptions; and average counts of each type of node per graph.

automatically [39] and then the boundaries are refined manually such that each clip corresponds to one *social situation*.

We developed a web-based annotation tool that allows human annotators to create graphs of arbitrary size by explicitly creating nodes and connecting them via a drag-and-drop interface. Two key points of our dataset are that each annotator: 1) creates an entire graph per clip, ensuring that each graph is *globally coherent* (i.e., the emotions, interactions, topics make sense when viewed together); and 2) annotates a complete movie, so that the graphs for *consecutive clips* in a movie are also coherent—this would not be possible if annotators simply annotated randomly-assigned clips from a movie. We provide details on annotation and dataset below.

3.1. Annotation Interface

Our annotation interface allows an annotator to view movie clips sequentially. For each clip, the annotator was asked to specify the scene and situation, write a natural language summary, and create a detailed graph of the situation, as depicted in Fig. 1. We describe each component of the annotation:

The **scene label** provides information about the location where the situation takes place, *e.g. office, theater, airport*.

The **situation label** corresponds to the high-level topic of the clip, and summarizes the social interactions that occur between characters, *e.g. robbery*, *wedding*.

The **description** provides a multi-sentence, natural language summary of what happens in the clip, based on video, dialog, and any additional information the annotator inferred about the situation.

The **graph** represents a human's understanding of a given situation. Our graphs feature 8 different types of nodes, with edges between them to indicate dependencies. We allow the annotator to choose the *directionality* of each edge. A graph consists of the following node types:

Character nodes represent the people in a scene. We provide a comprehensive list of character names obtained from IMDb[1], which the annotators can drag and drop onto the graph canvas.

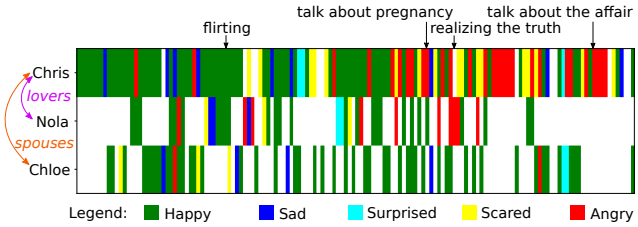

Figure 2: Emotional timelines of the three main characters in "Match Point." The emotions are correlated with situations and relationships between characters.

Attributes can be added to character nodes. The categories of attributes are: age, gender, ethnicity, profession, appearance, mental and emotional states.

Relationship nodes can link two or more characters. The relationships can refer to: family (*e.g. parent*, *spouse*), friendship/romance (*e.g. friend*, *lover*), or work (*e.g. boss*, *co-worker*). A relationship node can be tagged with a start/end token if it starts or ends in a given clip (*e.g.* the *spouse* relationship starts in a *wedding* clip). Otherwise, we assume that the characters were already in the relationship prior to the scene (*e.g.* already married).

Interaction nodes can be added to link two or more characters. Interactions can be either verbal (*e.g. suggests*, *warns*) or non-verbal (*e.g. hugs*, *sits near*). They can be directed (from one character to another, *e.g. A helps B*), or bidirectional if the interaction is symmetric (*e.g. A and B argue)*. A *summary interaction* captures the gist of several local interactions. Typically there is a single directed summary interaction from each character to the other (*e.g. argues*), while there may be many local ones (*e.g. asks, replies*).

Topic nodes can be added to interactions to add further details. For example, the interaction *suggests* may have the topic *to quit the job*.

Reason nodes can be added to interactions and attributes to provide motivations. For example, *apologizes* (interaction) can be linked to *he was late* (reason). Reasons can also be added to emotions: for example, *happy* (emotion) can be linked to *she got engaged* (reason). Reason nodes contain *inferred* common-sense information. See Table 2 for examples of topics and reasons.

Time stamp nodes ground the graph in the video clip, by providing the time interval in which an interaction or emotional state takes place (*e.g.* a character is *sad*, then (s)he becomes *happy*).

We also perform automatic face tracking, and ask annotators to assign a character name to each track (or mark as false positive). Thus, character nodes are grounded in videos.

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

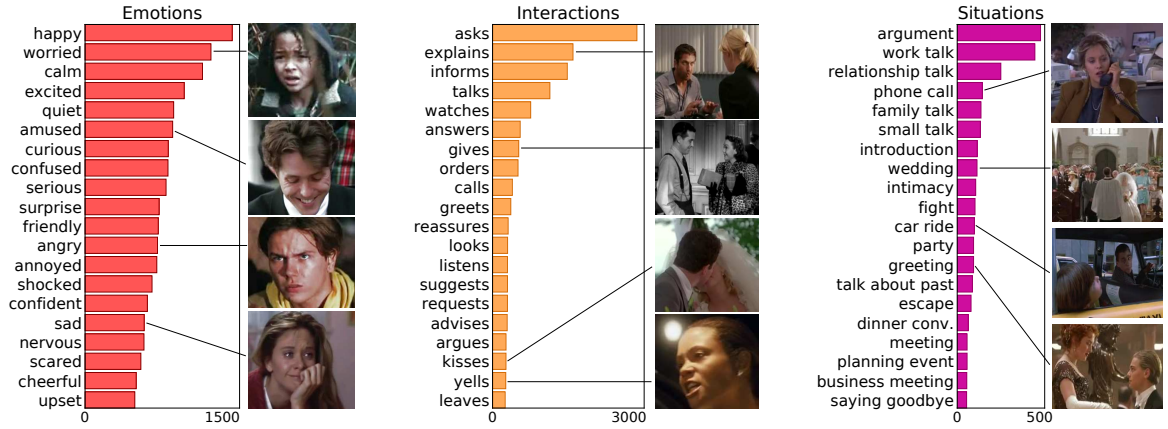

Figure 3: Distributions of the top 20 emotion attributes, interactions, and situations.

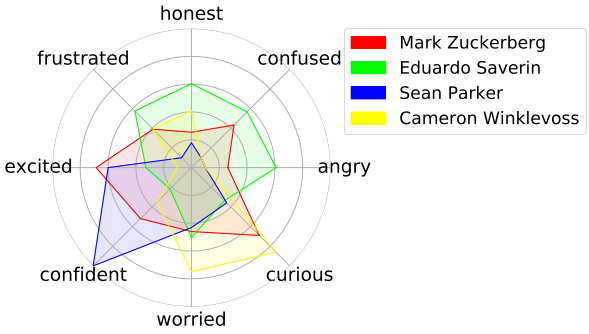

Figure 4: Emotional profiles from "The Social Network."

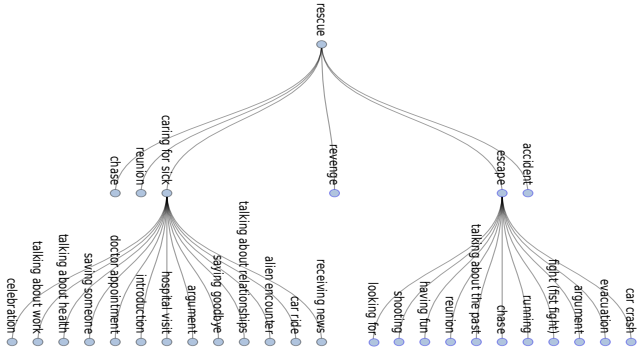

Figure 5: Flow of situations aggregated from all movies.

3.2. Data Collection Procedure

We hired workers via the freelance website Upwork. We worked closely with a small group of annotators, to ensure high-quality annotations. The workers went through a training phase in which they annotated the same set of clips according to an instruction manual. After gathering annotations, we also had a *cross-checking* phase, where annotators swapped movies and checked each others' work.

3.3. Dataset Statistics

Our dataset consists of 7637 annotated clips from 51 movies. Dataset statistics are shown in Table 1. The majority of the clips contain between 2 and 4 characters, and a graph has on average 13.8 attributes and 3.1 interactions. Fig. 3 shows the distributions of the top 20 emotion attributes, interactions, and situations. We show correlations between node types for a selected set of labels in Fig. 6, and the most common social aspects of scenes associated with the situation *party*, to showcase the insight offered by our dataset.

The dataset annotations allow us to follow a character throughout a movie. Fig. 2 shows the emotions experienced by the three main characters of the movie "Match Point," clip by clip. The emotions make sense when viewed in the context of the situations: when the characters flirt, they are happy; when they talk about problematic issues (pregnancy, the truth, the affair), they are angry. Fig. 4 shows the emotional profiles of characters from the movie "The Social Network," obtained by aggregating the characters' emotions over all clips.

In movies, like in real life, situations follow from other situations. In Fig. 5, we present a tree of situations rooted at *rescue*; this is essentially a knowledge graph that shows possible pairwise transitions between situations.

4. Situation Understanding Tasks

Graphs are an effective tool for capturing the gist of a situation, and are a structured alternative to free-form representations such as textual descriptions. We propose three tasks to demonstrate different aspects of situation understanding: 1) video clip retrieval using graphs as queries; 2) interaction sorting; and 3) reason prediction. In this section, we describe these tasks and propose models to tackle them.

4.1. Graph-Based Situation Retrieval

Here we aim to use graphs as queries to retrieve relevant clips from our dataset, where each clip consists of video and dialog. We assume our query is a graph $G = (\mathcal{V}, \mathcal{E})$ consisting of different types of nodes $v^{type} \in \mathcal{V}$ and edges between them. We use the notation v^{ch}, v^{att}, v^{rel}, v^{int}, v^{topic}, and v^{reason} to denote character, attribute, relationship, interaction, topic, and reason nodes. Character nodes

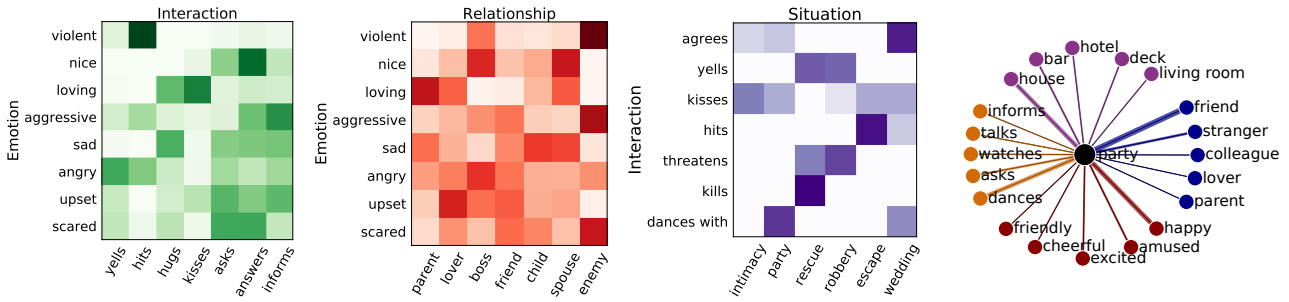

Figure 6: **Left:** Correlations between social aspects of scenes. **Right, clockwise from top:** The top-5 most common scenes, relationships, attributes, and interactions associated with the situation *party*.

Interaction: "asks"	
Topic	**Reason**
who she is	she is pretty
is this love at first sight	can't stop looking at her
if he is sober	he is the driver
about the speech	he's the best man
about wedding gifts list	he needs to buy one
for help	he is late again
if he is for bride or groom	to determine seating area
for time alone	to think

Table 2: Topics and reasons associated with the interaction "asks" in the movie "Four Weddings and a Funeral."

are the backbone of the graph: all other nodes (except for topics and reasons) link to at least one character node. To ease notation, we consider the scene and situation labels as disconnected nodes in the graph, v^{sc} and v^{si}, respectively.

To perform retrieval, we aim to learn a real-valued function $F_\theta(M, G)$ that scores the similarity between a movie clip M and the query graph G, where F_θ should score the highest for the most relevant clip(s). At test time, we are interested in retrieving the clip with the highest similarity with G. We design F_θ to exploit the structure of the graph and evaluate it against a clip in a semantically meaningful way. In particular, we reason about the *alignment* between character nodes in the graph and face/person tracks in the video. Given an alignment, we score attributes, interactions, and other nodes accordingly.

Each clip M typically contains several video shots. We automatically parse each clip to obtain face tracks in each shot, and cluster tracks with similar faces across shots. To model interactions and relationships, we extend the face detection boxes to create full-body person tracks. We represent each cluster c_j with a feature vector $\mathbf{x}_j$, and a pair of clusters (c_j, c_k) with a feature vector $\mathbf{x}_{jk}$. Global information about the clip is captured with a feature vector $\mathbf{x}_{scene}$. Additional details are provided in Sec. 4.1.2.

We define a random variable $\mathbf{z} = (z_1, \ldots, z_N)$ which reasons about the alignment between character nodes v_i^{ch} and face clusters, $z_i \in \{1, \ldots, K\}$. Here, N is the number of character nodes in the query G, and K is the number of face clusters in the clip. We restrict $\mathbf{z}$ to map different nodes to different clusters, resulting in all permutations $\mathbf{z} \in P(K, N)$. In practice, $N = 5$ and $K = 7$.

We define a function that scores a graph in the video clip given an alignment $\mathbf{z}$ as follows:

$$\begin{aligned} F_\theta(M, G, \mathbf{z}) &= \phi_{sc}(v^{sc}) + \phi_{si}(v^{si}) \\ &+ \sum_i \left(\phi_{ch}(v_i^{ch}, z_i) + \phi_{att}(\mathcal{V}_i^{att}, z_i) \right) \\ &+ \sum_{i,j} \phi_{int}(\mathcal{V}_{ij}^{int}, z_i, z_j) + \sum_{i,j} \phi_{rel}(\mathcal{V}_{ij}^{rel}, z_i, z_j). \end{aligned} \quad (1)$$

The set of attributes associated with character i is $\mathcal{V}_i^{att} = \{v_k^{att} : (i,k) \in \mathcal{E}\}$ and the set of interactions between a pair of characters (i, j) is $\mathcal{V}_{ij}^{int} = \{v_k^{int} : (i,k), (k,j) \in \mathcal{E}\}$, where all edges are directed. The set of relationships is defined similarly. Here, ϕ are potential functions which score components of the graph in the clip. Each ϕ also depends on the clip M and learned parameters θ, which we omit for convenience of notation.

We now describe each type of potential in more detail. To form the query using the graph, we embed each node label using word embeddings. For nodes that contain phrases, we mean-pool over the words to get a fixed length representation $\mathbf{a}^{type}$ (where $type$ is att, int, etc.). In our case, $\mathbf{a}^{type} \in \mathbb{R}^{100}$ (GloVe [32]). We learn two linear embeddings for each type, W_g^{type} for query node labels and W_m^{type} for observations, and score them in a joint space. We share W_g across all node types to prevent overfitting.

Video-Based Potentials. The attribute unary potential computes the cosine similarity between node embeddings and visual features:

$$\phi_{att}(\mathcal{V}_i^{att}, z_i) = \left\langle W_g \sum_{v_k \in \mathcal{V}_i^{att}} \mathbf{a}_k^{att}, W_m^{att} \mathbf{x}_{z_i}^{att} \right\rangle. \quad (2)$$

A similar potential is used to score the scene v^{sc} and situation v^{si} labels with video feature $\mathbf{x}_{scene}$ (but does not depend on

$\mathbf{z}$). Furthermore, we score pairwise dependencies as:

$$\phi_{type}(\mathcal{V}_{ij}^{type}, z_i, z_j) = \left\langle W_g \sum_{v_k \in \mathcal{V}_{ij}^{type}} \mathbf{a}_k^{type}, W_m^{type} \mathbf{x}_{z_i z_j} \right\rangle \quad (3)$$

with $type \in \{rel, int\}$.

Scoring Dialog. To truly understand a situation, we need to consider not only visual cues, but also dialog. For this, we learn a function Q to score a query G with dialog D as:

$$Q(D, G) = \sum_{v_k \in \mathcal{V}} \sum_i \max_j ((W_g \mathbf{a}_{k,i})^T (W_d \mathbf{x}_{d_j})), \quad (4)$$

where $\mathbf{a}_{k,i}$ is the GloVe embedding of the i^{th} word in node v_k, and $\mathbf{x}_{d_j}$ is the embedding of the j^{th} dialog word. This finds the best matching word in the dialog for each word in the graph, and computes similarity by summing across all graph words. We initialize the matrices W_g and W_d to identity, because GloVe vectors already capture relevant semantic information. To take into account both video and dialog, we perform late fusion of video and dialog scores (see Sec. 5.1).

Person Identification. To classify each face cluster as one of the characters, we harvest character and actor images from IMDb. We fine-tune a VGG-16 [30] network on these images, combined with our video face crops, using a triplet loss (i.e., minimizing the Euclidean distance between embeddings of two positive examples wrt a negative pair). To compute $\phi_{ch}(v_i^{ch}, z_i)$, we find the embedding distance between the face track and each movie character, and convert it into a probability. For details, see Suppl. Mat. D.

4.1.1 Learning and Inference

Learning. Our training data consists of tuples $(G_n, M_n, \mathbf{z}_n)$: for each graph we have an associated clip and ground-truth alignment to face clusters. We learn the parameters of F_θ using the max-margin ranking loss:

$$\mathcal{L}_\theta = \sum_{(n,n')} \max(0, 1 - (F_\theta(G_n, M_n, \mathbf{z}_n) - F_\theta(G_n, M_{n'}, \mathbf{z}_{n'}))), \quad (5)$$

where n' is an index of a negative example for G_n. In practice, we sample three classes of negatives: 1) clips from other movies (different characters, therefore easy negatives); 2) different clips from the same movie (medium difficulty); or 3) the same clip with different alignments $\mathbf{z}_{n'}$ (same characters, aligned with the clip incorrectly, therefore hard negatives). We train the dialog model $Q(D, G)$ similarly, with a max-margin ranking loss that does not involve $\mathbf{z}$. We use the Adam optimizer [17] with learning rate 0.0003.

Inference. We perform an exhaustive search over all clips and alignments to retrieve the most similar clip for the query graph G:

$$M^* = \arg\max_n \left(\max_{\mathbf{z}} F_\theta(G, M_n, \mathbf{z}) \right). \quad (6)$$

4.1.2 Implementation Details

Video Features. To obtain a holistic video representation, we process every fifth frame of the video using the Hybrid1365-VGG model [46] and extract `pool5` features. We mean pool over space and time to obtain one representation $\mathbf{x}_{scene} \in \mathbb{R}^{512}$ for the entire clip. For each face cluster, we compute age and gender predictions [20] (Eq. 2, $\mathbf{x}_{z_i}^{age}$, $\mathbf{x}_{z_i}^{gen}$) and extract features from another CNN trained to predict emotions [21] (Eq. 2, $\mathbf{x}_{z_i}^{att}$). This allows us to score unary terms involving attributes.

We extend the face detections to obtain person detections and tracks that are used to score pairwise terms. We represent each person track by pooling features of spatio-temporal regions in which the person appears. Specifically, $\mathbf{x}_{z_i z_j}$ (Eq. 3) is computed by stacking such person track features $[\mathbf{x}_{z_i}^p; \mathbf{x}_{z_j}^p]$. Note that ordered stacking maintains edge directions ($v_i^{ch} \rightarrow v^{int,rel} \rightarrow v_j^{ch}$).

Text. We evaluate two representations for text modalities: (i) TF·IDF [24], where we use the logarithmic form; and (ii) GloVe [32] word embeddings. Similar to [40], scoring the dialogs with TF·IDF involves representing the graph query and dialog text as sparse vectors ($\mathbb{R}^{|\text{vocab}|}$) and computing their cosine similarity. We explore two pooling strategies with word embeddings: 1) *max-sum* (Eq. 4); and 2) *max-sum · idf*, which weighs words based on rarity.

4.2. Interaction Ordering

Predicting probable future interactions on the basis of past interactions, their topics, and the social context is a challenging task. We evaluate interaction understanding via the proxy task of learning to *sort* a set of interactions into a plausible order (Table 5). We present a toy task wherein we take the interactions between a pair of characters, and train an RNN to *choose interactions* sequentially from the set, in the order in which they would likely occur. We represent an interaction and corresponding topic by the concatenation of their GloVe embeddings, with an additional digit appended to indicate the direction of the interaction. We train an attention-based decoder RNN to regress interaction representations: at each time step, it outputs a vector that should be close to the embedding of the interaction at that step in the sequence. We use a single-layer GRU [7], and condition on a 100-d context vector formed by applying linear layers on the situation, scene, relationship, and attribute embeddings. We zero-mask one interaction from the input set at each time step, to ensure that the model does not select the same interaction multiple times. Masking is done with teacher-forcing during training, and with the model's predictions at test time. For details, see Suppl. Mat. B.

4.3. Reason Prediction

Given information about the scene in the form of attributes of each character, their relationship, and an interac-

tion in which they are engaging, we aim to predict plausible reasons for why the interaction took place. Scene and situation labels are also used, to provide global context.

As in previous tasks, we first represent the relevant nodes by their GloVe embeddings. The characters are identity agnostic and are represented as a weighted combination of their attributes. We encode individual components of the sub-graph (scene, situation, interaction, relationship) through linear layers and learn a 100-d context vector.

Our decoder is a single-layer GRU with 100 hidden units that conditions on the scene description (context vector), and produces a reason, word by word. As is standard, the decoder sees the previous word and context vector at each time step to generate the next word. To obtain some variability during sampling, we set the temperature to 0.6. We train the model end-to-end on the train and val sets (leaving out a few samples to choose a checkpoint qualitatively), and evaluate on test. Please refer to Suppl. Mat. C for details.

5. Experimental Results

The *MovieGraphs* dataset is split into *train*, *val*, and *test* sets with a 10:2:3 ratio of clips (see Table 1), and no overlapping movies. We learn model parameters on *train*, choose checkpoints on *val*, and present final evaluation on *test*.

Face Clustering and Person Identification. On average, our clips have 9.2 valid face tracks which form 2.1 ground-truth clusters. For face clustering, we obtain a weighted clustering purity of 75.8%, which is reasonable, as we do not filter background characters or false-positive tracks. Person identification (ID) for a large number of movies spanning many decades is hard, due to the differences between IMDb gallery images and video face tracks. We obtain a track-level identification accuracy of 43.7% vs. chance at 13.2%. We present details in Suppl. Mat. D.

5.1. Graph-based Retrieval

All retrieval results are shown in Table 3. Similar to image-text retrieval (*e.g.* Flickr8k [12]), we use the following metrics: median rank and recall at K (1, 5, 10). Unless mentioned otherwise, we assume that the entire graph is used as part of the query. The first two rows show the performance of a random retrieval model that may or may not know the source movie.

Description Retrieval. Our first experiment evaluates the similarity between graphs and clip descriptions. We use the three models described in Sec. 4.1.2: TF·IDF, *max-sum*, and *max-sum* · *idf*. We consistently obtain median rank 1 (Table 3, rows 3-5), possibly due to descriptive topics and reasons, and character names that help localize the scene well (see Suppl. Mat. A for an ablation study on node types).

Dialog Retrieval. In our second experiment, we aim to retrieve a relevant clip based on dialog, given a graph. This is considerably harder, as many elements of the graph are visual (*e.g. kisses*) or inferred from the conversation (*e.g. encourages*). We evaluate dialog retrieval with the same models used for descriptions. Here, GloVe models (rows 7, 8) perform worse than TF·IDF (row 6) achieving med.-R 10 vs 6. We believe that this is because the embeddings for several classes of words are quite similar, and confuse the model.

	Method	PersonID		TEST			
		CL	ID	R@1	R@5	R@10	med.-R
1	random, movie unkn.	-	-	0.1	0.3	0.7	764
2	random, movie known	-	-	0.7	3.3	6.6	78
	DESCRIPTION						
3	TF·IDF	-	-	61.6	83.8	89.7	1
4	GloVe, max-sum	-	-	62.1	81.3	87.2	1
5	GloVe, idf · max-sum	-	-	61.3	81.6	86.9	1
	DIALOG						
6	TF·IDF	-	-	31.8	49.8	57.2	6
7	GloVe, max-sum	-	-	28.0	42.4	50.2	10
8	GloVe, idf · max-sum	-	-	28.7	43.1	50.2	10
	MOVIE CLIP						
9	sc	-	-	1.1	4.3	7.7	141.5
10	sc, si	-	-	1.0	5.4	8.7	140
11	sc, si, att	pr	pr	2.2	9.4	15.5	84
12	sc, si, att, rel, int	pr	pr	2.7	10.9	18.9	59
13	sc, si, att, rel, int	pr	gt	7.7	28.8	44.9	13
14	sc, si, att, rel, int	gt	gt	13.0	37.4	50.4	10
15	sc, si, att, rel, int, dlg	pr	pr	31.6	50.4	56.6	5
16	sc, si, att, rel, int, dlg	gt	gt	40.4	62.1	71.1	3

Table 3: Retrieval results when using the graph as a query. *dlg* refers to dialog. For PersonID, CL and ID indicate clustering and identification; *gt* denotes ground-truth, and *pr* denotes predictions.

Movie Clip Retrieval. Our third experiment evaluates the impact of visual modalities. Note that if the query consists only of scene or situation labels, there are multiple clips that are potential matches. Nevertheless, starting from a random median rank of 764, we are able to improve the rank to 141.5 (row 9) with the scene label only, and 140 with scene and situation labels (row 10). Directly mapping high-level situations to visual cues is challenging.

Similar to the way characters help localize descriptions and dialogs, person identification helps localization in the visual modality. If our query consists only of characters, in the best case scenario of using ground-truth (gt) clustering and ID, we obtain a median rank of 17. Our predicted (pr) clustering works quite well, and obtains median rank 19. Owing to the difficulty of person ID, using pr clustering and ID pushes the median rank to 69.

Rows 9-16 present an ablation of graph components. We start with the scene, situation, attributes, and characters (rows 9-11) as part of our query graphs. Including interactions (+topics) and relationships improves the rank from 84 to 59 (row 12). In a scenario with gt clusters and ID, we see a large improvement in med.-R from 59 to 10 (rows 13, 14).

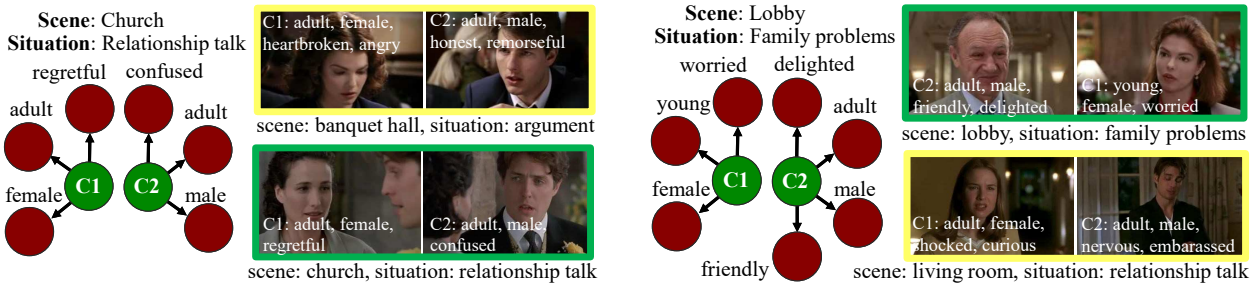

Figure 7: Identity agnostic sub-graph queries and the top-2 retrieved clips, which are from different movies. We search for video clips that have overall similarity with respect to scene and situation, and also character attributes and emotions. The yellow boxes indicate results that are quite similar in meaning to the query, and the green boxes indicate ground-truth.

Fully Sorted Accuracy	Longest Common Subsequence
40.5% (27%)	0.74 (0.67)

Table 4: Performance of our interaction sorting approach on the test set. The number in (·) is random chance.

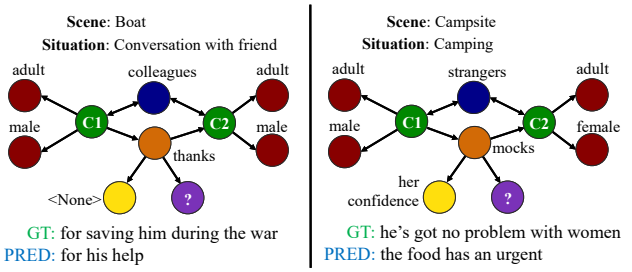

Figure 8: Example sub-graphs with GT and predicted reasons. The left one is scored *Very relevant*, and the right *Not relevant*. However, the model's mistake of relating camping with food is reasonable.

Late Fusion. We combine video and dialog cues using late fusion of the scores from the models used in rows 8 and 12/14, and see a large increase in performance in both pr-pr and gt-gt settings (rows 15, 16). This points to the benefits offered by dialog in our model.

Qualitative Results. We present an example of sub-graph retrieval in Fig. 7. Even with small queries (sub-graphs) and identity agnostic retrieval, we obtain interesting results.

5.2. Interaction Ordering

We measure ordering performance using two metrics: (i) the model's accuracy at predicting complete sequences; and (ii) the length of the longest common subsequence (LCS) between the ground-truth and prediction. Quantitative results are shown in Table 4. Table 5 shows qualitative examples of the orderings predicted by our model. The first two sequences are correctly sorted by our model, while the third is a failure case. However, even in the failure case, interactions 2, 3, 4, and 5 are in the correct order (longest common subsequence), and the entire sequence is plausible.

5.3. Reason Prediction

An interaction can have several distinct and relevant reasons, making automatic scoring using captioning metrics hard. We ask 10 AMT workers to score 100 sub-graphs and their predicted reasons as: *Very relevant*, *Semi-relevant*, and *Not relevant*. Fig. 8 shows two examples, along with their GT and predicted reasons. We are able to obtain a clear verdict (6 or more annotators agree) on 72 sub-graphs: 11 samples are rated very relevant, while 10 more are semi-relevant.

GT	Pred	Dir.	Interaction + [Topic]
1	1	→	asks [why she's crying]
2	2	←	explains to [why she is sad]
3	3	→	comforts
1	1	→	waits for [to end the audition]
2	2	←	informs [audition went bad]
3	3	→	suggests [they have a drink]
4	4	←	agrees
1	2	→	explains [it's great to know who he wants]
2	3	→	advises [to go visit her]
3	1	←	expresses doubt [she may not like him]
4	6	→	encourages
5	4	←	thanks [for the advice]
6	5	←	announces [he is going to her]

Table 5: Qualitative results for ordering interactions. Each interaction is shown with its topic in brackets. The interactions are listed in their ground-truth order, and the predicted sequence is shown in the "Pred" column, where the numbers represent the order in which the interaction is predicted. The → indicates that C1 initiates the interaction with C2, and ← the reverse.

6. Conclusion

In this work, we focused on understanding human-centric situations in videos. We introduced the *MovieGraphs* dataset, that contains rich annotations of everyday social situations in the form of graphs. Our graphs capture people's interactions, emotions, and motivations, many of which must be inferred from a combination of visual cues and dialog. We performed various statistical analyses of our dataset and proposed three tasks to benchmark situation understanding: graph-based video retrieval, interaction understanding via ordering, and reason prediction. We proposed models for each of the tasks, that point to their successes and challenges.

Acknowledgments. Supported by the DARPA Explainable AI (XAI) program, NSERC, MERL, and Comcast. We thank NVIDIA for their donation of GPUs. We thank Relu Patrascu for infrastructure support, and we thank the Upwork annotators.

References

[1] CMU's HERB Robotic Platform, http://www.cmu.edu/herb-robot/.

[2] Hollywood2: Human Actions and Scenes Dataset, http://www.di.ens.fr/~laptev/actions/hollywood2/.

[3] Large Scale Movie Description Challenge, https://sites.google.com/site/describingmovies/.

[4] Microsoft's Tay, https://twitter.com/tayandyou.

[5] J.-B. Alayrac, P. Bojanowski, N. Agrawal, I. Laptev, J. Sivic, and S. Lacoste-Julien. Unsupervised Learning from Narrated Instruction Videos. In *CVPR*, 2016.

[6] P. Bojanowski, F. Bach, I. Laptev, J. Ponce, C. Schmid, and J. Sivic. Finding Actors and Actions in Movies. In *ICCV*, 2013.

[7] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[8] L. Ding and A. Yilmaz. Learning Relations among Movie Characters: A Social Network Perspective. In *ECCV*, 2010.

[9] M. Eichner, M. Marin-Jimenez, A. Zisserman, and V. Ferrari. 2D Articulated Human Pose Estimation and Retrieval in (Almost) Unconstrained Still Images. *IJCV*, 99(2):190–214, 2012.

[10] M. Everingham, J. Sivic, and A. Zisserman. "Hello! My name is ... Buffy" – Automatic Naming of Characters in TV Video. In *BMVC*, 2006.

[11] A. Fathi, J. K. Hodgins, and J. M. Rehg. Social Interactions: A First-Person Perspective. In *CVPR*, 2012.

[12] M. Hodosh, P. Young, and J. Hockenmaier. Framing Image Description as a Ranking Task: Data, Models and Evaluation Metrics. *Journal of Artificial Intelligence Research*, 47:853–899, 2013.

[13] Y. Jang, Y. Song, Y. Yu, Y. Kim, and G. Kim. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. *arxiv.org/pdf/1704.04497.pdf*, 2017.

[14] N. Jaques, Y. L. Kim, and R. W. Picard. Personality, Attitudes, and Bonding in Conversations. In *Proc. of Intelligent Virtual Agents*, 2016.

[15] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei. Image Retrieval using Scene Graphs. In *CVPR*, 2015.

[16] K.-M. Kim, M.-O. Heo, S.-H. Choi, and B.-T. Zhang. DeepStory: Video Story QA by Deep Embedded Memory Networks. *arxiv.org/pdf/1707.00836.pdf*, 2017.

[17] D. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980*, 2014.

[18] J. Kory Westlund, J. J. Lee, L. Plummer, F. Faridi, J. Gray, M. Berlin, H. Quintus-Bosz, R. Hartmann, M. Hess, S. Dyer, K. dos Santos, S. Örn Adhalgeirsson, G. Gordon, S. Spaulding, M. Martinez, M. Das, M. Archie, S. Jeong, and C. Breazeal. Tega: A Social Robot. In *International Conference on Human-Robot Interaction*, 2016.

[19] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning Realistic Human Actions from Movies. In *CVPR*, 2008.

[20] G. Levi and T. Hassner. Age and Gender Classification Using Convolutional Neural Networks. In *CVPR Workshop*, 2015.

[21] G. Levi and T. Hassner. Emotion Recognition in the Wild via Convolutional Neural Networks and Mapped Binary Patterns. In *ICMI*, 2015.

[22] J. Li, M. Galley, C. Brockett, G. P. Spithourakis, J. Gao, and B. Dolan. A Persona-Based Neural Conversation Model. *arXiv:1603.06155*, 2016.

[23] R. Li, M. Tapaswi, R. Liao, J. Jia, R. Urtasun, and S. Fidler. Situation Recognition with Graph Neural Networks. In *ICCV*, 2017.

[24] C. D. Manning, P. Raghavan, and H. Schutze. Scoring, Term Weighting, and the Vector Space Model. In *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[25] M. J. Marin-Jimenez, A. Zisserman, M. Eichner, and V. Ferrari. Detecting People Looking at Each Other in Videos. *IJCV*, 106(3):282–296, 2014.

[26] M. Marszałek, I. Laptev, and C. Schmid. Actions in Context. In *CVPR*, 2009.

[27] M. J. Matarič. Socially Assistive Robotics: Human Augmentation vs. Automation. *Science Robotics*, 2(4), 2017.

[28] J. Mun, P. H. Seo, and I. J. B. Han. MarioQA: Answering Questions by Watching Gameplay Videos. *arxiv.org/pdf/1612.01669.pdf*, 2017.

[29] S.-B. Park, K.-J. Oh, and G. Jo. Social Network Analysis in a Movie Using Character-Net. In *Multimedia Tools and Applications*, 2011.

[30] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep Face Recognition. In *BMVC*, 2015.

[31] A. Patron-Perez, M. Marszalek, I. D. Reid, and A. Zisserman. Structured Learning of Human Interactions in TV Shows. *PAMI*, 34(12):2441–2453, 2012.

[32] J. Pennington, R. Socher, and C. D. Manning. Glove: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[33] V. Ramanathan, A. Joulin, P. Liang, and L. Fei-Fei. Linking People in Videos with "Their" Names Using Coreference Resolution. In *ECCV*, 2014.

[34] A. Rohrbach, M. Rohrbach, N. Tandon, and B. Schiele. A Dataset for Movie Description. In *CVPR*, 2015.

[35] O. Sener, A. Zamir, S. Savarese, and A. Saxena. Unsupervised Semantic Parsing of Video Collections. *arXiv:1506.08438*, 2015.

[36] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *ECCV*, 2016.

[37] M. Tapaswi, M. Bauml, and R. Stiefelhagen. Book2Movie: Aligning Video Scenes with Book chapters. In *CVPR*, 2015.

[38] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. "Knock! Knock! Who is it?" Probabilistic Person Identification in TV series. In *CVPR*, 2012.

[39] M. Tapaswi, M. Bäuml, and R. Stiefelhagen. StoryGraphs: Visualizing Character Interactions as a Timeline. In *CVPR*, 2014.

[40] M. Tapaswi, Y. Zhu, R. Stiefelhagen, A. Torralba, R. Urtasun, and S. Fidler. MovieQA: Understanding Stories in Movies through Question-Answering. In *CVPR*, 2016.

[41] A. Torabi, C. Pal, H. Larochelle, and A. Courville. Using Descriptive Video Services To Create a Large Data Source For Video Annotation Research. *arXiv:1503.01070*, 2015.

[42] O. Vinyals and Q. Le. A Neural Conversational Model. *arXiv:1506.05869*, 2015.

[43] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei. Scene Graph Generation by Iterative Message Passing. In *CVPR*, 2017.

[44] M. Yatskar, V. Ordonez, L. Zettlemoyer, and A. Farhadi. Commonly Uncommon: Semantic Sparsity in Situation Recognition. In *CVPR*, 2017.

[45] M. Yatskar, L. Zettlemoyer, and A. Farhadi. Situation Recognition: Visual Semantic Role Labeling for Image Understanding. In *CVPR*, 2016.

[46] B. Zhou, A. Khosla, A. Lapedriza, A. Torralba, and A. Oliva. Places: An Image Database for Deep Scene Understanding. *arXiv:1610.02055*, 2016.

[47] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning Books and Movies: Towards Story-like Visual Explanations by Watching Movies and Reading Books. In *ICCV*, 2015.