SUPPLEMENTARY MATERIAL
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`

Supplementary Material Overview

In this document, we provide additional details about our dataset, models, and results. It is structured as follows:

- We present an ablative study of TF·IDF-based retrieval, to examine the discriminative power of each node type (Sec. A).
- We present details of our interaction ordering model, and more qualitative results (Sec. B).
- We present details of our reason prediction model, and more qualitative results (Sec. C).
- We provide details of person detection, clustering, and identification (Sec. D).
- We describe the annotation interface used to collect the data (Sec. E).
- We present example annotations for one movie from the MovieGraphs dataset (Sec. F), as well as additional dataset statistics:
  - The distributions of the number of characters, interactions, and relationships per clip;
  - The top-20 relationships and scenes.
- We provide additional examples of the MovieGraphs dataset (Sec. G), and show visualizations of:
  - The emotional profiles and timelines of the main characters from several movies;
  - Emotions of characters on both sides of interactions/relationships;
  - A rooted graph of situations;
  - Examples of interaction annotations.

A. TF·IDF Ablative Study

In this section, we present an ablative study of TF·IDF graph → description and graph → dialog retrieval. In order to gauge the relative discriminative power of each node type for retrieval, we perform the retrieval experiment considering only information in certain node types.

Node Types	**R@1**	**R@5**	**R@10**	**medR**
All Node Types	61.6	83.8	89.8	1
All $\setminus$ { Sc., Sit. }	60.8	82.1	88.0	1
All $\setminus$ { Sc., Sit., Char.}	51.0	67.8	74.2	1
Scene	3.1	9.9	12.9	460
Situation	5.0	13.5	18.3	382
Character	19.0	45.3	58.7	7
Attribute	3.7	9.0	12.7	393
Interaction, Summary	12.8	24.6	29.9	66
Interaction	6.2	12.8	16.2	327
Summary	7.2	15.8	20.8	264
Relationship	0.6	1.8	2.6	704
Topic	35.5	51.4	57.3	5
Reason	24.5	38.0	42.4	26

Table 1: TF·IDF Graph-Description Retrieval Ablation Study

Node Types	**R@1**	**R@5**	**R@10**	**medR**
All Node Types	31.8	49.8	57.2	6
All $\setminus$ { Sc., Sit. }	31.9	49.4	57.1	6
All $\setminus$ { Sc., Sit., Char.}	31.5	46.2	51.5	8
Scene	0.7	2.2	3.1	718
Situation	1.2	3.5	4.8	627
Character	6.6	17.4	23.2	58
Attribute	1.0	3.4	5.7	603
Interaction, Summary	3.3	6.5	8.4	580
Interaction	1.8	4.1	5.8	621
Summary	1.6	4.1	5.8	656
Relationship	0.3	1.3	2.0	765
Topic	28.7	42.6	47.9	15
Reason	17.4	28.7	32.9	122

Table 2: TF·IDF Graph-Dialog Retrieval Ablation Study

We present results on the test set, evaluated using recall @ $\{1, 5, 10\}$ and median rank. Table 1 shows the results for graph → description retrieval, and Table 2 shows the graph → dialog retrieval results. In both cases, we find that character, topic, and reason nodes are the most discriminative for localizing the correct clip. This aligns with the intuition that the most relevant information involves who is in the clip, and what the details (topics and reasons) of their interactions are.

B. Interaction Ordering Details

In this section, we present details of our interaction ordering model, which uses an attention-based RNN to select interactions sequentially from an input set to form a plausible order.

Dataset Creation for Interaction Ordering. We extract all sequences of interactions that occur between each pair of characters in each clip. We consider only interactions that have time stamps, and sort them first based on start time, and then by end time to break ties.

Interaction Sequence Encoding. Each training example represents a sequence of interactions between a pair of characters, and consists of: 1) a situation label; 2) a scene label; 3) the relationship(s) between the two characters; 4) the attributes of each character; and 5) a sequence of N interactions (with associated topics).

Consider two characters C_1 and C_2, with attributes $\mathcal{V}^{att}_{C_1}$ and $\mathcal{V}^{att}_{C_2}$, respectively. Let the scene and situation labels be v^{sc} and v^{si}, respectively, and the relationships between C_1 and C_2 be $v^{rel}_{C_1 \to C_2}$ and $v^{rel}_{C_1 \leftarrow C_2}$, respectively.

Each interaction $v^{int}_{C_1,C_2}$ with topic $v^{top}_{C_1,C_2}$ is represented by the concatenation of the corresponding GloVe embeddings, $[\mathbf{a}^{int}_{C_1,C_2}; \mathbf{a}^{top}_{C_1,C_2}]$, with an additional digit appended to indicate the direction of the interaction: 1 for $C_1 \to C_2$, -1 for $C_1 \leftarrow C_2$, and 0 for $C_1 \leftrightarrow C_2$. We denote the interaction representation at step i in the sequence by $\mathbf{x}_i$, so that a sequence of length N is denoted by $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$.

Context Encoding. We create a global context vector (passed to the decoder RNN at each time step) that incorporates situation, scene, relationship, and attribute information. We restrict attributes to be of age and gender types, and compute attribute encodings as follows:

$$h_{C_1} = W_{C_1} \sum_k \alpha_k \mathbf{a}^{att}_{C_1 k}, \quad h_{C_2} = W_{C_2} \sum_l \alpha_l \mathbf{a}^{att}_{C_2 l} \tag{1}$$

where the attention weights α_k and α_l are computed using a two-layer MLP.

We encode the relationships between C_1 and C_2 as follows:

$$h_{r_1} = W_{r_1} \mathbf{a}^{rel}_{C_1 \to C_2}, \quad h_{r_2} = W_{r_2} \mathbf{a}^{rel}_{C_1 \leftarrow C_2} \tag{2}$$

Bidirectional relationships are treated as separate relationships in both directions, $C_1 \to C_2$ and $C_1 \leftarrow C_2$.

We encode the scene and situation with linear layers on the corresponding GloVe representations:

$$h_{sc} = W_{sc} \mathbf{a}^{sc} \tag{3}$$
$$h_{si} = W_{si} \mathbf{a}^{si} \tag{4}$$

Finally, we combine the encoded components to form the global context vector:

$$z = h_{C_1} + h_{C_2} + h_{r_1} + h_{r_2} + h_{sc} + h_{si} \tag{5}$$

Attention-Based Decoder RNN. We use a single-layer Gated Recurrent Unit (GRU) RNN to select items sequentially from an input set of interactions. At each time step t, we compute a local context vector by attending to the input elements:

$$c^{(t)} = \sum_{i=1}^{N} \alpha_i^{(t)} \mathbf{x}_i \tag{6}$$

where

$$\alpha^{(t)} = \text{softmax}(s^{(t)}) \tag{7}$$

and

$$s_i^{(t)} = (h^{(t-1)})^T \mathbf{x}_i \,. \tag{8}$$

The input to the GRU at time t consists of 1) an input element $\mathbf{x}^{(t)}$ (described below for the train and test settings); 2) the local context $c^{(t)}$; and 3) the global context z:

$$[\mathbf{x}^{(t)}; c^{(t)}; z] \,. \tag{9}$$

The output is computed as a linear transformation on the hidden state:

$$o^{(t)} = W_o h^{(t)} \tag{10}$$

We score each interaction from the input set by computing the inner product between the output $o^{(t)}$ and each of the interaction representations $\mathbf{x}_i$. The model is trained end-to-end with cross-entropy loss on these scores. At test time, the input element with the highest score is chosen at each time step, and is masked out from the input set so that it is not selected again in future steps. At training time, we use teacher forcing (i.e., choosing the correct ground-truth interaction to be passed forward to the next step, regardless of which interaction scored the highest) 50% of the time, and the model's own predictions 50% of the time.

Results. In Table 3 we show additional qualitative interaction-ordering results. In examples (a)-(d) the predicted order matches the ground-truth, while in examples (e)-(i) the predicted order does not match. However, even though the orders presented in examples (e)-(i) are not identical to the ground-truth orders, they are still plausible.

Ex.	GT	Pred	Dir.	Interaction + [Topic]
a)	1	1	$\rightarrow$	asks [who was on the phone]
	2	2	$\leftarrow$	informs [a friend called]
	3	3	$\rightarrow$	orders [to go care for the child]
b)	1	1	$\rightarrow$	calls to
	2	2	$\leftarrow$	ignores
	3	3	$\rightarrow$	offers [treat]
c)	1	1	$\rightarrow$	asks [will they ever be happy]
	2	2	$\leftarrow$	answers [that they are happy]
	3	3	$\leftarrow$	suggests [they borrow her old crib]
	4	4	$\rightarrow$	hushes
d)	1	1	$\rightarrow$	asks about [identity]
	2	2	$\leftarrow$	yells at
	3	3	$\rightarrow$	yells [to get out]
	4	4	$\leftarrow$	accuses [that he pretends to be a millionaire]
e)	1	4	$\rightarrow$	confesses to [he did not murder anyone]
	2	5	$\rightarrow$	explains to [he went to jail for video pirating]
	3	3	$\leftarrow$	reproaches [they thought he was a murderer]
	4	1	$\leftarrow$	asks [about his gun]
	5	2	$\rightarrow$	shows [he doesn't have a gun]
f)	1	4	$\rightarrow$	begs [not to go]
	2	2	$\leftarrow$	apologizes
	3	5	$\leftarrow$	explains to [she's retiring]
	4	3	$\leftarrow$	encourages [to be strong]
	5	1	$\rightarrow$	calls after
g)	1	1	$\rightarrow$	orders [to shut up]
	2	2	$\leftarrow$	accuses [of assault and battery]
	3	4	$\rightarrow$	threatens [to beat him up]
	4	5	$\rightarrow$	warns [he must make it up to his friend]
	5	3	$\rightarrow$	admits [he doesn't like to yell]
h)	1	1	$\rightarrow$	informs [they lost all the stuff]
	2	4	$\leftarrow$	asks [if ex-worker set the fire]
	3	3	$\rightarrow$	explains to [the cause of her husband's death]
	4	2	$\leftarrow$	informs [that they have to succeed]
	5	5	$\rightarrow$	reassures [they won't let them win]
i)	1	3	$\rightarrow$	asks [why her eyes look old]
	2	4	$\leftarrow$	jokes [his eyes look kind]
	3	2	$\leftarrow$	explains to [she woke up early]
	4	1	$\rightarrow$	interrupts [to stop her from clearing his plate]

Table 3: Additional qualitative results for interaction ordering. The GT column shows the ground-truth order of interactions, the Pred column shows the order in which the interactions were chosen by the model, and the Dir column shows the direction of the interaction between the characters.

C. Reason Prediction Details

In this section, we present details of our reason prediction model. In particular, we build a model that encodes the context of the interaction and generates plausible reasons using a decoder RNN.

Context Encoding. The context is comprised of: (i) a pair of characters C_1, C_2, represented by their attributes $\mathcal{V}^{att}_{C_1}$ and $\mathcal{V}^{att}_{C_2}$ (actual names don't tell us anything and do not matter); (ii) the interaction $v^{int}_{C_1,C_2}$ for which we wish to predict the reason; (iii) the relationship $v^{rel}_{C_1,C_2}$ between the characters; (iv) the scene v^{sc} and situation v^{si} labels; (v) and optionally, a topic $v^{top}_{C_1,C_2}$ node associated with the interaction.

In particular, we restrict attributes to age and gender (as others, such as emotions, were found to have little influence). The characters are represented by a weighted combination of attributes:

$$h_{C_1} = \sum_k \alpha_k \mathbf{a}^{att}_{C_1 k}, \quad h_{C_2} = \sum_l \alpha_l \mathbf{a}^{att}_{C_2 l} \tag{11}$$

where the attention weights α_k, α_l are learned using a two-layer MLP. $\mathbf{a}_.$ corresponds to the 100-d GloVe embedding for each node.

In our graphs, interactions and relationships can take three possible directions: $C_1 \rightarrow C_2$ (*e.g.* parent), $C_1 \leftarrow C_2$ (*e.g.* child), and $C_1 \leftrightarrow C_2$ (*e.g.* friends). We first encode a direction $C_e \rightarrow C_f$:

$$h_d^{e\rightarrow f} = W_{C_d} \cdot h_{C_e} + W_{C_r} \cdot h_{C_f} \,, \tag{12}$$

where W_{C_d} corresponds to the parameters for the *doer* and W_{C_r} the *receiver*. Interactions and relationships are encoded as:

$$\begin{aligned} h_i^{e\rightarrow f} &= h_d^{e\rightarrow f} + W_i \cdot (\mathbf{a}^{int}_{C_1,C_2} + \mathbf{a}^{top}_{C_1,C_2}) \,, & (13)\\ h_r^{e\rightarrow f} &= h_d^{e\rightarrow f} + W_r \cdot \mathbf{a}^{rel}_{C_1,C_2} \,, & (14)\\ h_{i|r}^{e\leftrightarrow f} &= (h_{i|r}^{e\rightarrow f} + h_{i|r}^{e\leftarrow f})/2 \,. & (15) \end{aligned}$$

The final context vector is a linear combination:

$$h_c = W_p \mathbf{a}^{sc} + W_s \mathbf{a}^{si} + h_i + h_r \,. \tag{16}$$

Decoder RNN. We adopt a Gated Recurrent Unit RNN [3] with a 100-d hidden state as our decoder. The context is fed in at each time-step along with the previous sampled word (or `START` token). We use a temperature sampling scheme to generate variability in the predicted reasons.

Results. Fig. 1 shows several more examples of context graphs and ground-truth and predicted reasons. Each row shows the results for test set samples annotated as *Not relevant* (row 1), *Semi-relevant* (row 2) and *Very relevant* (row 3). Note that the results in the last row, while not predicting the ground-truth reason, are very plausible, affirming the difficulty of this task.

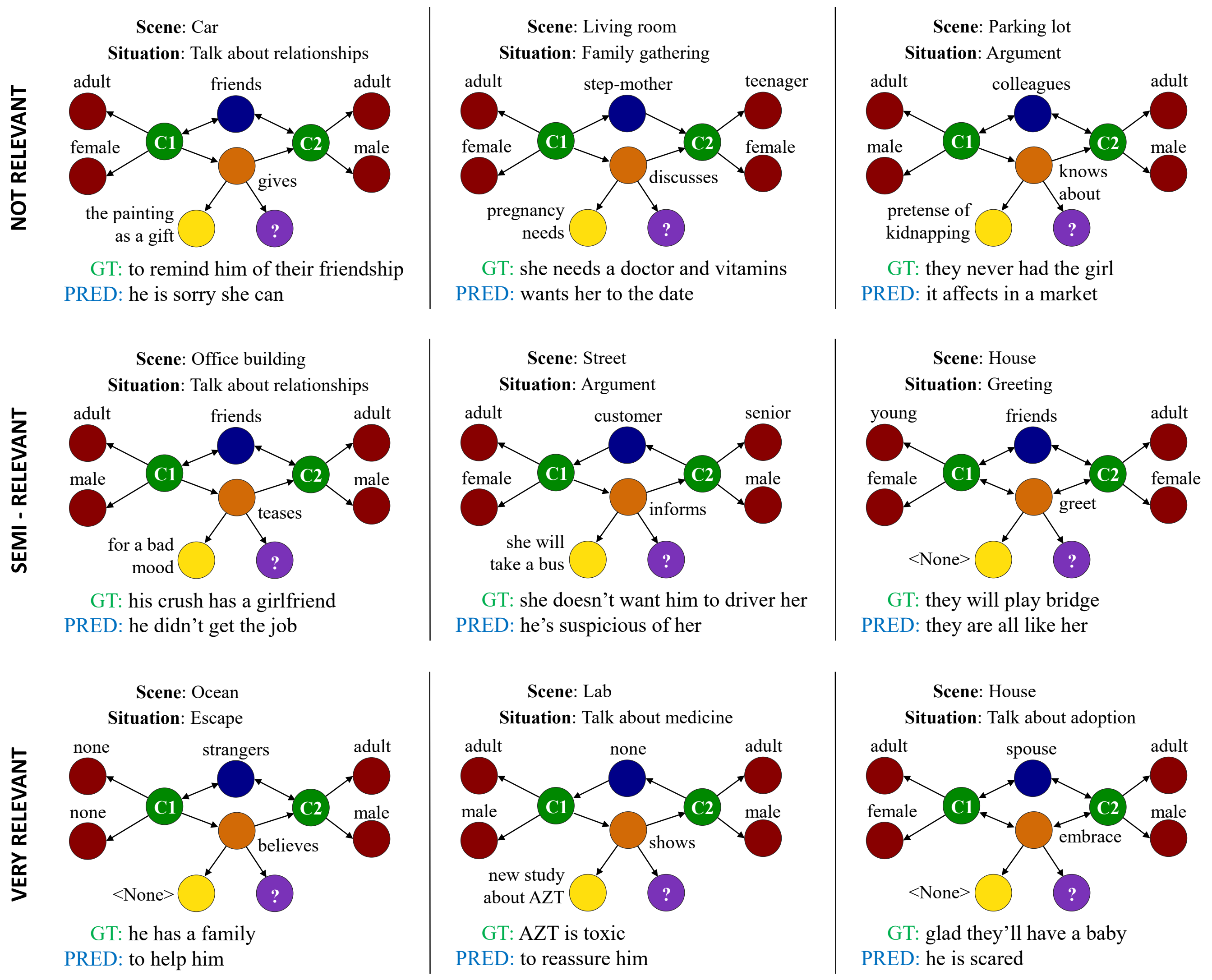


Figure 1: Reason prediction results on the test set, grouped by the three evaluation labels.

D. Person Detection, Clustering, and Identification

Data Collection for Face ID. We grounded each character in the graph with all the face tracks in the clip (obtained with OpenFace [2]). As there are on average 1740 face tracks per movie, 9.6 per clip, and over 88K in our full dataset (prior to removing false positives and track switches), annotating the tracks directly would have been time consuming. Instead, we first linked face tracks that belong to the same person in a clip by performing hierarchical agglomerative clustering on upper body color histogram features (a character wears the same clothing during a clip) corresponding to the face tracks. The annotator is then asked to assign character names (from those found in the graph) to clusters in the clip. In cases where clustering is wrong, we break the cluster into face tracks, and the annotator labels each track.

Face Detection and Tracking. Face tracks are extracted using the OpenFace tracker [2]. The output of the tracker are face detections in each frame, which we need to post-process into face tracks. To match detections from one frame to the detections in subsequent frames, we compute the IoU (intersection over union) between each possible pair of detections from the different frames. We then compute the optimal matching of pairs that maximizes the IoU between pairs using the Hungarian algorithm. Since detections might be spurious, we allow for gaps of up to 5 frames in which the face might not be detected in the same face track and we discard every face track that lasts less than 10 frames. We note that the quality of our face tracker is primarily limited by the face detector, which occasionally results in missing face tracks and false positives. Furthermore, while we found our post-processing heuristics to work reasonably in practice, there are some track switches grouped together. On average, we obtain 66 false positive face tracks and track switches per movie (corresponding to less than 4% of all tracks).

Face Features. We compute face embeddings for the purpose of character identification and face track clustering. Our model builds upon the VGG-Face model of [6]. We extend it by (i) reducing the dimensionality of the features from 4096 to 128 with a linear projection layer (*i.e.*, a fully connected layer without non-linearity), and (ii) making them have a unit norm. We initialize our model with the weights of [6] and train it using a triplet loss as described in [7] with a margin $m = 0.2$ but without mining hard negatives due to limited computational resources.

To train our model we use our own dataset, comprised of images from face tracks and cast pictures from IMDb. Face track identities are annotated in the dataset, while IMDb pictures have associated identity metadata. Furthermore, we filter IMDb pictures by running a standard face detection algorithm [9] and including only those with a single face detection. Each triplet in our dataset is formed by a face track anchor, a positive example randomly sampled either from matching face tracks or IMDb pictures, and a negative example also from either non-matching IMDb pictures or face tracks. Since a face track contains many images, we randomly select one of them to represent a face track example in a triplet during training. At test time, a face track is represented by the average features of its images. We feed face crops into our model instead of full images, expanding the face detection bounding box by a factor of 2 to also include clothing and hair. The context was found to be especially helpful for track clustering. Expanded crops are resized to a fixed resolution of 224x224 using bilinear interpolation.

Face Track Clustering. We perform face track clustering in each movie scene to group together face tracks belonging to the same characters. To perform clustering, we use face features for each of the face tracks and compute pairwise distances between them. We then apply hierarchical agglomerative clustering with a cut-off at threshold $t = 0.75$, empirically chosen to minimize the OCI metric on the validation set.

We evaluate our clustering performance in the test set with three different metrics: (i) cluster purity, indicating how many clusters capture a single identity, (ii) weighted cluster purity, in which the purity of a cluster is weighted by the number of face tracks in the cluster, and (iii) operator clicks index (OCI) [5], indicating the number of face tracks in wrong clusters plus the total number of clusters obtained. This last metric seeks to evaluate the cost of annotating face tracks given a clustering. While evaluating our method, we discard wrong face tracks or face tracks that switch faces. The performance of our method is shown in Table 4.

Metric	Score
Cluster Purity	75.7%
Weighted Purity	75.8%
OCI	6.4

Table 4: Clustering performance

Character Identification. Here we explore the task of assigning character identities to face tracks. We extract face features for cast pictures with a single face detection and compute pairwise distances with face track images. We then rank the images by distance to the face tracks and assign character identity probabilities to each face track. We evaluate our assignments with accuracy in Table 5. We keep the top-15 characters that appear in at least 10 graphs, as otherwise, movies (including all extras and background) can have characters (and actors) that often do not even have pictures on IMDb.

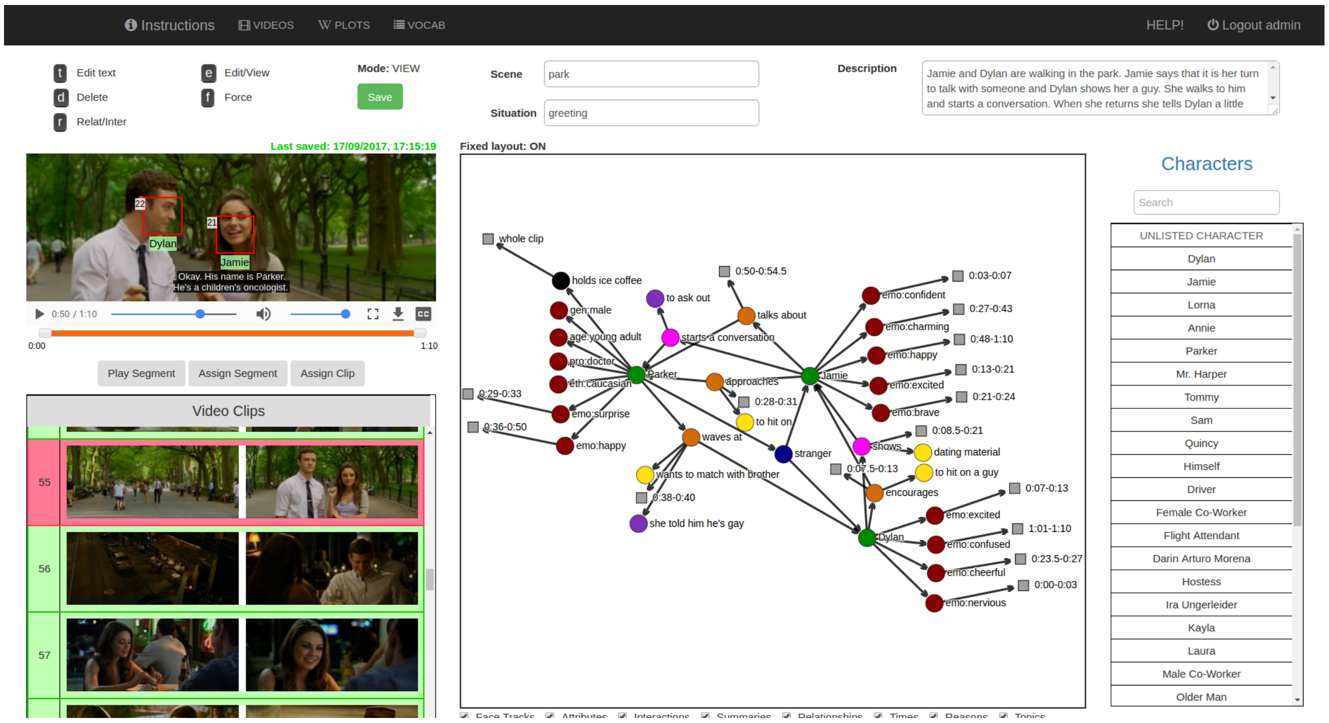
Metric	Ours	Random
Accuracy	43.7%	13.2%

Table 5: Character identification

E. Annotation Interface

In this section, we describe our data collection procedure. We developed a custom web-based annotation interface (Fig. 2) to support our data collection process. An annotator first selects a movie clip from the list on the left-hand side; the clip to be annotated then plays at the top. For each clip, the annotator must annotate four elements: 1) a scene label; 2) a situation label; 3) a natural language description; and 4) a graph. The graph canvas is in the center of the page; text boxes for scene and situation labels, as well as for the description, are above. In order to create a graph, an annotator starts by adding *characters*. The list of characters (from the IMDb cast of the movie) is shown on the right; clicking a character name from the list creates the corresponding character node in the graph canvas. An unlisted character can be added by choosing *unlisted character* and then naming it. All other node types (e.g., attributes, interactions, relationships, topics, reasons, and time stamps) are added directly in the graph canvas. We provided initial vocabularies for scenes and situations which annotators could select from, but they were also allowed to add custom items to the lists. The interactions, topics, and reasons were free-form, and we encouraged the annotators to be concise.

Workflow. Each annotator watched a movie from start to finish, annotating each clip in order. This ensured that each



Scene: park

Situation: greeting

Description: Jamie and Dylan are walking in the park. Jamie says that it is her turn to talk with someone and Dylan shows her a guy. She walks to him and starts a conversation. When she returns she tells Dylan a little

Okay. His name is Parker.
He's a children's oncologist.

Figure 2: The MovieGraphs annotation interface provides four separate areas where annotators type the scene label, the situation label, and the natural language description, and where they create the graph. On the left, they select a clip from a sequence of clips, and on the right, they select characters from a cast list (obtained from IMDb).

annotation contained data inferred from the movie up until that point in time, and allowed annotators to gain insight into the reasons behind each character's actions. The typical workflow of the annotator was to watch the clip and first write the description in natural language, then provide the scene and situation label, and finally create the graph. Good data requires proper identification of the scene and situation, a description thorough enough that people reading it can understand what happens in the clip, and a graph detailed enough that reading it conveys the gist of the clip, as well.

Training Annotators on Upwork. We hired annotators through the freelance service Upwork. Therefore, our setup had to overcome some unique challenges, the main one being uniformity: the annotators have to agree on the level of detail they put in the graph. To support data quality, we trained annotators on a common set of clips, and continued monitoring the annotations.

Dividing Movies into Clips. To obtain the clips for annotation, we automatically split each movie into scenes [8] and then grouped the scenes into clips manually, such that each clip corresponds to one coherent social situation (e.g., *party*). As some situations were longer than others, our clips vary in length from around 20 seconds to 2 minutes, with an average length of 44 seconds.

F. MovieGraphs Examples and Statistics

We show examples of the graph annotations for the movie "Jerry Maguire" in Fig. 3. We sample nine graphs from various points in the movie, to show the progression of the story.

Fig. 4 shows the top 20 relationships and scenes across all movies. The distribution of attribute types is shown in Fig. 5. We also present the distributions of the number of nodes of different types in each clip: the number of characters per clip is shown in Fig. 6; the number of interactions in Fig. 7; and the number of relationships in Fig. 8.

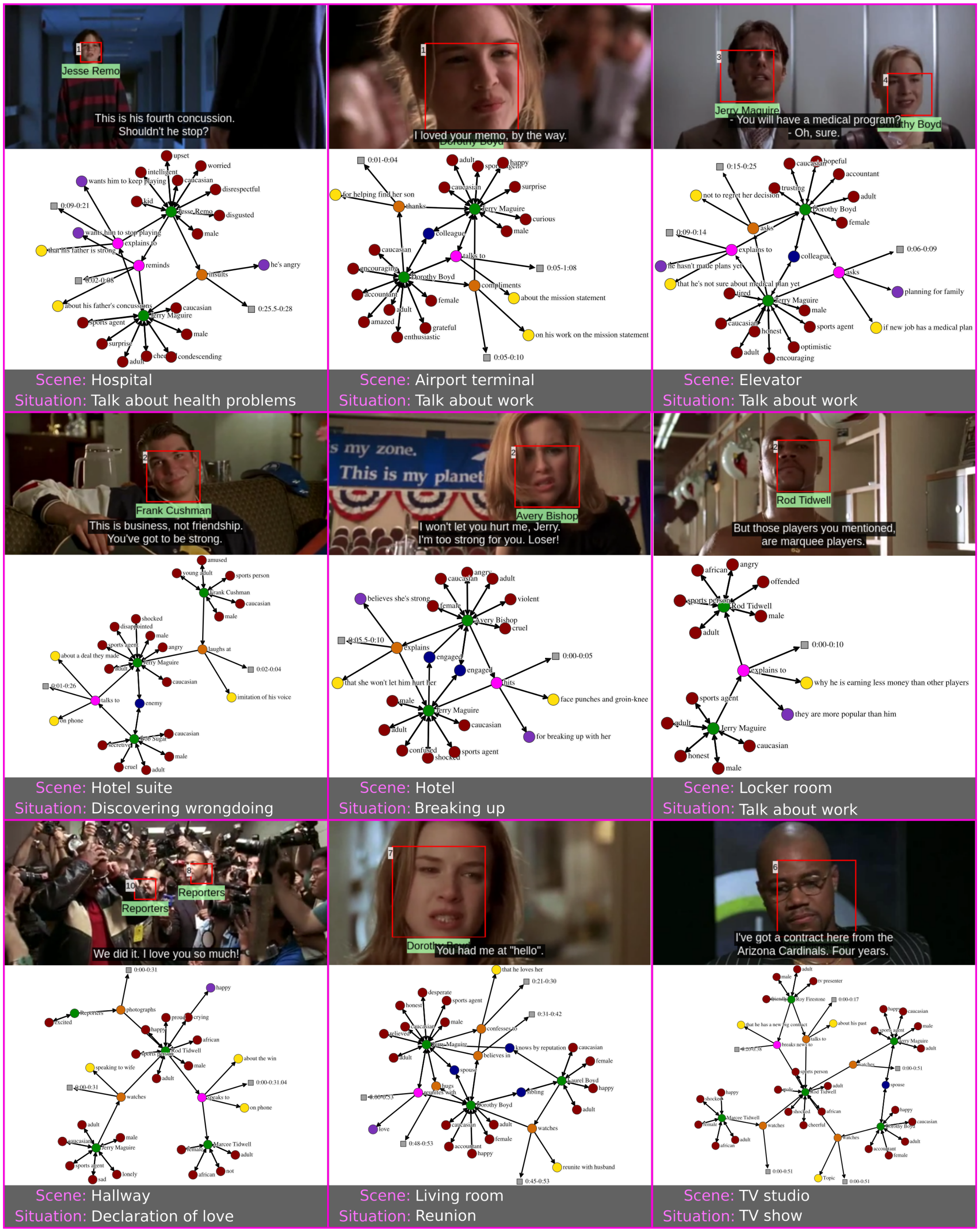

Jesse Remo
This is his fourth concussion. Shouldn't he stop?
Scene: Hospital
Situation: Talk about health problems
I loved your memo, by the way.
Scene: Airport terminal
Situation: Talk about work
Jerry Maguire
- You will have a medical program?
- Oh, sure.
Scene: Elevator
Situation: Talk about work
Frank Cushman
This is business, not friendship. You've got to be strong.
Scene: Hotel suite
Situation: Discovering wrongdoing
Avery Bishop
I won't let you hurt me, Jerry. I'm too strong for you. Loser!
Scene: Hotel
Situation: Breaking up
Rod Tidwell
But those players you mentioned, are marquee players.
Scene: Locker room
Situation: Talk about work
Reporters
We did it. I love you so much!
Scene: Hallway
Situation: Declaration of love
You had me at "hello".
Scene: Living room
Situation: Reunion
I've got a contract here from the Arizona Cardinals. Four years.
Scene: TV studio
Situation: TV show

Figure 3: Example annotations of various clips throughout the movie "Jerry Maguire," showing scenes, situations, and graphs. Each clip is also annotated with a natural language description (not shown due to space constraints).

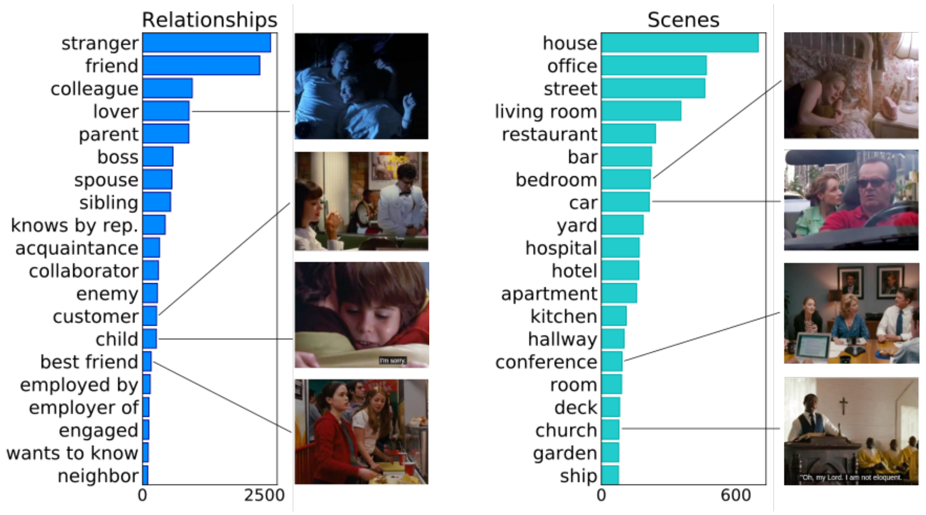


Figure 4: Distributions of the top 20 relationships and scenes.

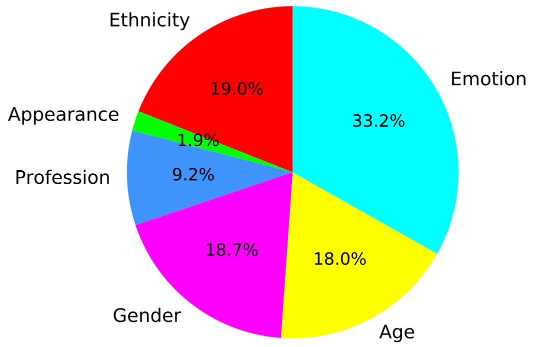


Figure 5: The distribution of attribute types.

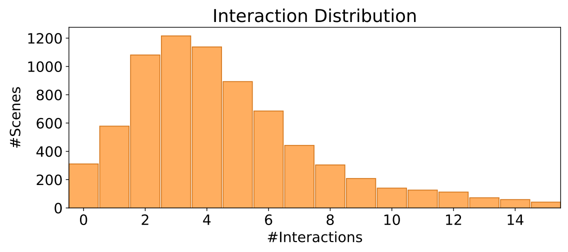


Figure 7: The distribution of the number of interactions per clip, over all movies.

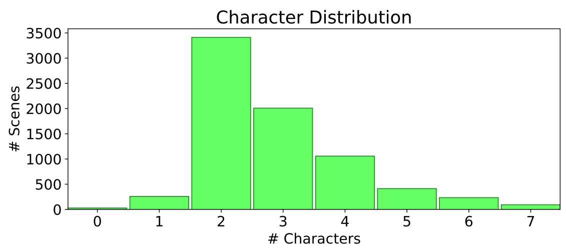


Figure 6: The distribution of the number of characters per clip, over all movies.

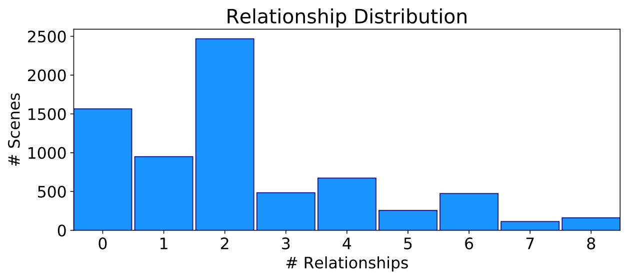


Figure 8: The distribution of the number of relationships per clip, over all movies.

G. More Examples

Character Emotional Profiles. We show the emotional profiles of the main characters in several movies, based on the emotions annotated for each character across all the clips in a movie. We see that in the movie "Four Weddings and a Funeral" (Fig. 9, left), characters are mostly sad, nervous (Charles), happy (Carrie), worried (Matthew) and excited (Tom), while in the movie "As Good As It Gets" (Fig. 9, right), the characters are often upset or grateful (Simon), embarrassed or angry (Melvin), happy (Carol), and angry or surprised (Frank).

Emotion Timelines. As in real life, characters in movies experience many emotions whose progression during the story we show in Fig. 10. We used the six basic emotions (angry, happy, sad, scared, surprised, and disgusted) described in [4]. We mapped the various words used to annotate emotions in the graphs to these six emotions based on [1], with some manual additions. Where a character was annotated to have several emotions in the same clip, we took the mode of these emotions. The emotions are color-coded, as shown in the legend for our figures.

We then correlated the emotional progression of the main characters in each movie with situations and relationships. In Fig. 10, we show the progression of emotions of the main characters in the movies "The Lost Weekend," "As Good as It Gets", and "The Social Network." For example, Fig. 10 (top diagram) shows the emotional timeline of four characters from the movie "The Lost Weekend." We show that Don, his brother Wick, and Don's girlfriend Helen plan a weekend trip to help Don break his alcohol addiction, but he is angry because he doesn't want to go. Don, still angry, goes to drink in a pub, then becomes sad when he is rejected by society. He continues drinking, and becomes scared during his alcoholic delirium episode. Eventually, he and Helen engage in a motivational conversation, and both end up happy. The timeline shows that Don is a troubled, angry character, while Helen, Wick, and Nat (the bartender) are calming influences.

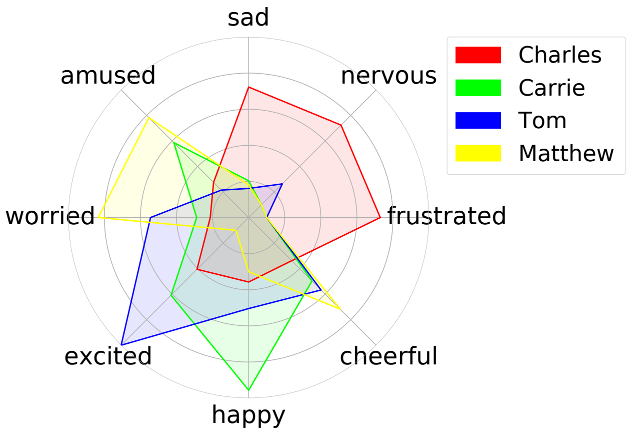
Character-Character Emotion Distributions. In many cases, an interaction is associated with different emotions for the "doer" and the "receiver." We show the distribution of emotions felt by characters on the giving and receiving ends of interactions (Fig. 11) and relationships (Fig. 12).

As shown in the top panel of the Fig. 11, for the interaction *attacks*, the attacker (Person 1) is often violent, angry, and aggressive, while the person being attacked (Person 2) is often scared and confused. The emotions of a person who begs (Fig. 11, middle panel) are also different (e.g. desperate, scared, helpless) from the emotions of the person on the other end of the interaction, who is often angry, calm, or compassionate. Fig. 11 (bottom panel) also shows that a person who *forgives* is forgiving and happy, while the person who is *forgiven* is often apologetic, happy, and grateful.

People also have different emotions depending on the relationship between them, as shown in Fig. 12. For example, a grandparent is often happy, while the grandchild is often scared (top panel); a mistress is often worried, and sometimes humiliated, while her lover is worried and sneaky (middle panel); and a nanny is often compassionate, while the child is often sad (bottom panel).

Rooted Situation Graph. We can also see how situations follow from one another. Fig. 13 shows possible pairwise transitions between situations, starting from the situation "date."

Interaction Examples. Fig. 14 shows example annotations of interactions, to showcase the fact that dialog is important for inferring many interactions, in addition to visual cues.



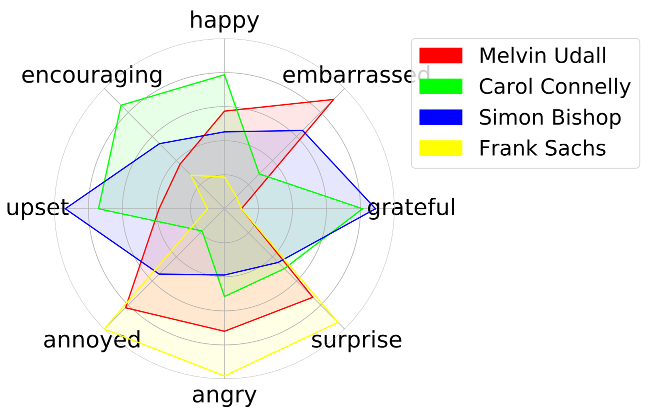
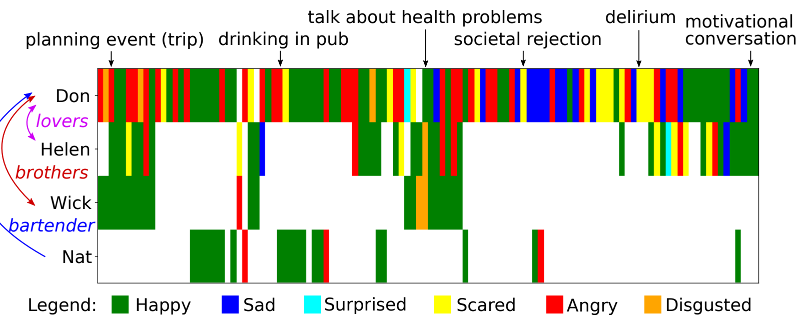
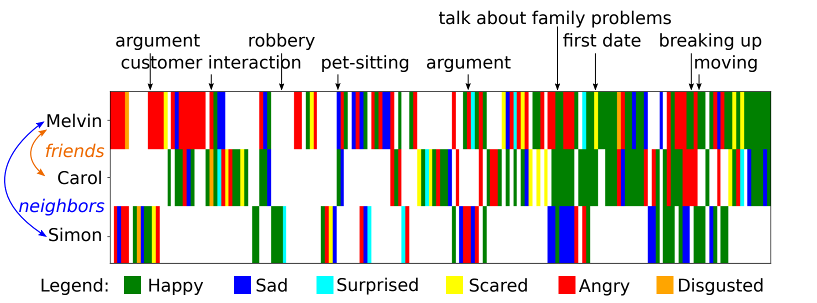


Figure 9: Emotional profiles of the main characters in "Four Weddings and a Funeral" (left) and "As Good as It Gets" (right). These profiles were obtained by aggregating the emotions of these characters over the whole movie.





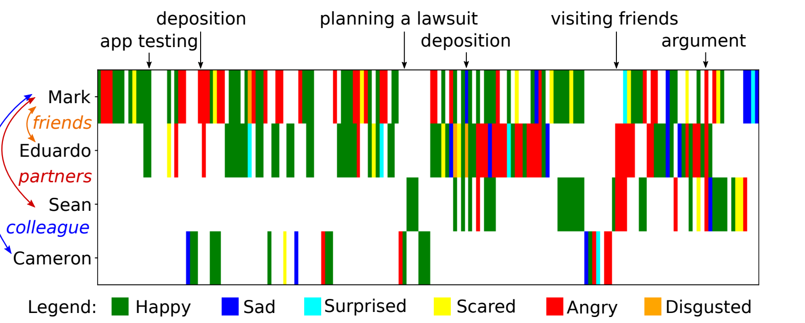


Figure 10: Emotional timelines for the main characters in "The Lost Weekend" (top), "As Good as It Gets" (middle), and "The Social Network" (bottom). The emotional timelines are correlated with situations (shown with arrows above each diagram) and relationships between characters (shown with arrows between names on the left). Wherever a character does not appear in a clip, the space is white.

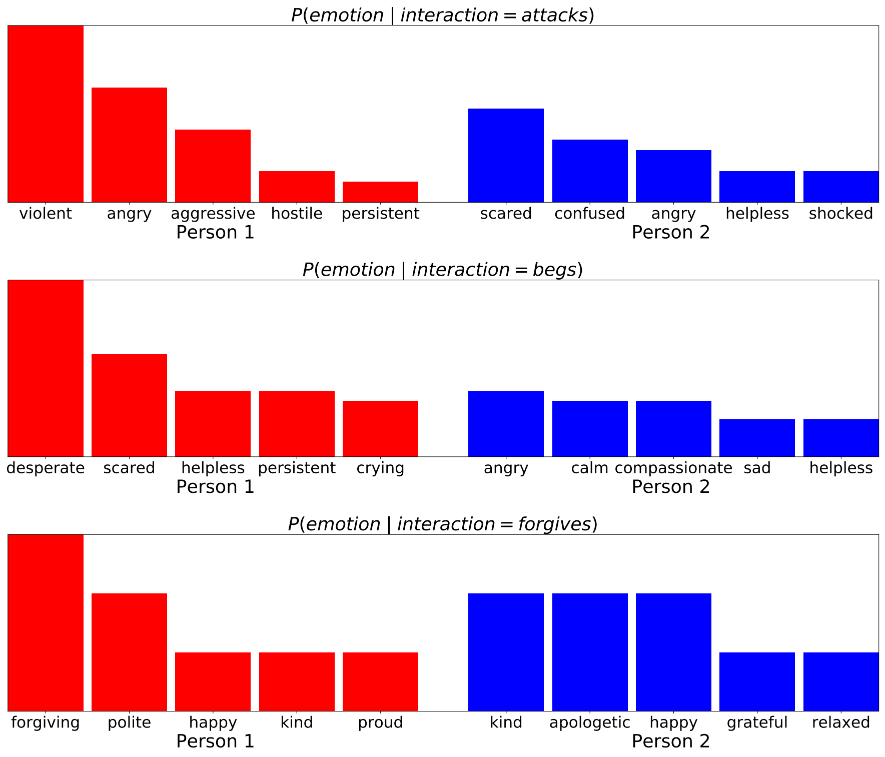


Figure 11: Emotions of characters on either side of an interaction. In each case, Person 1 directs the interaction toward Person 2 (e.g. in the top example, Person 1 *attacks*, Person 2 *is being attacked*).

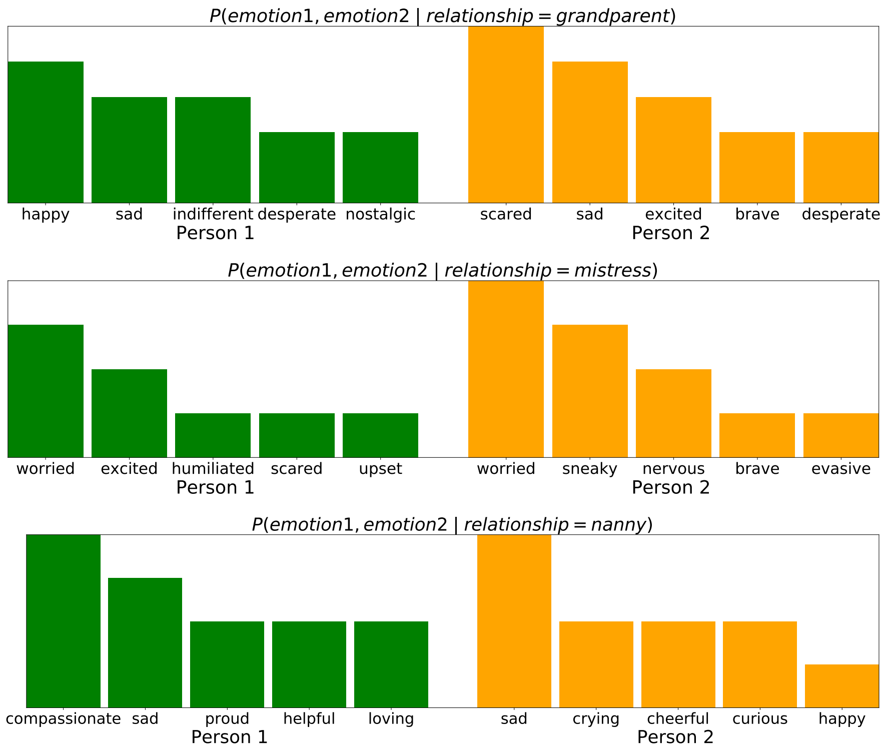


Figure 12: Emotions of characters on either side of a relationship. Each relationship is directed; in these examples, Person 1 is the grandparent, mistress, and nanny, while Person 2 is the grandchild, lover, and child, respectively.

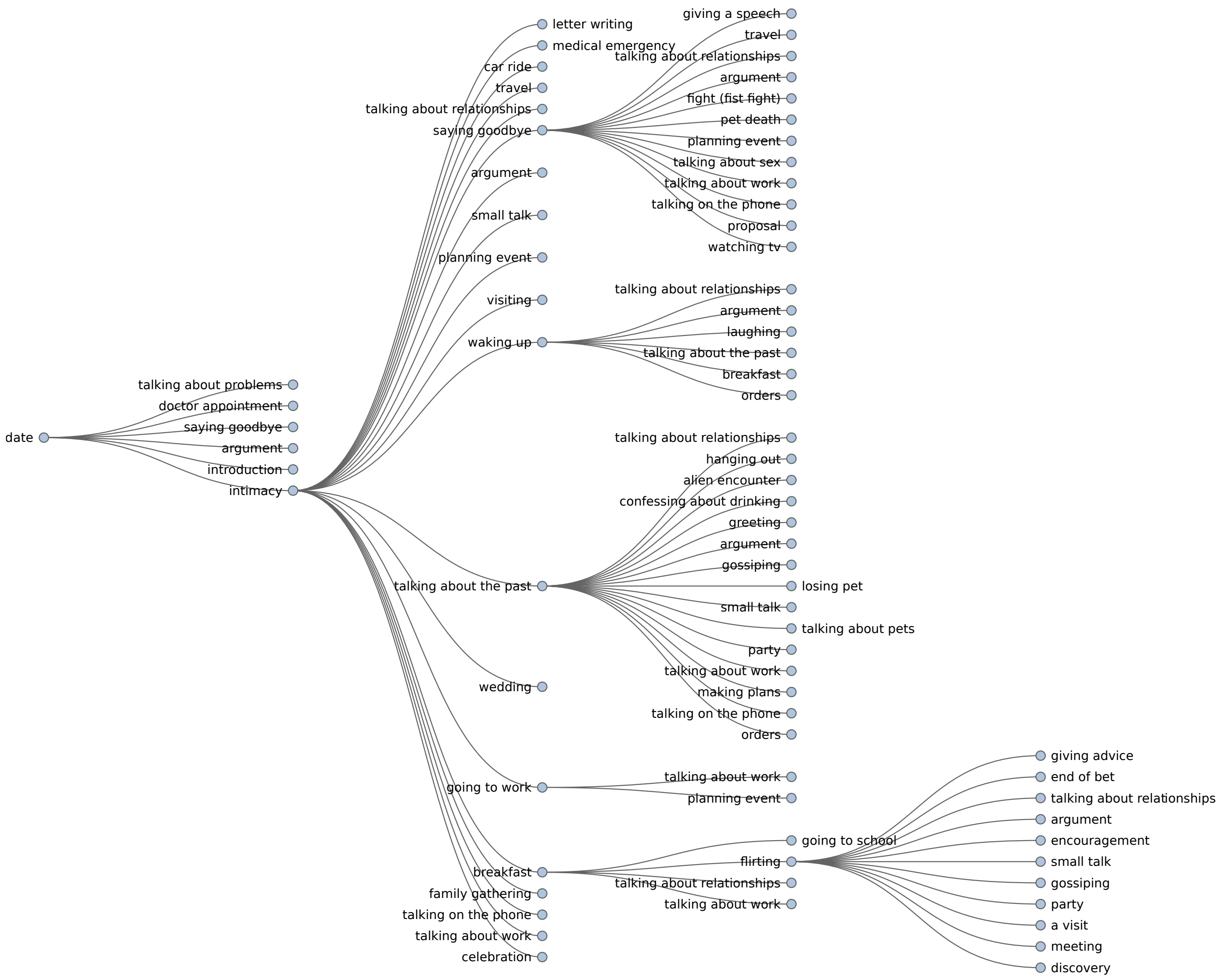
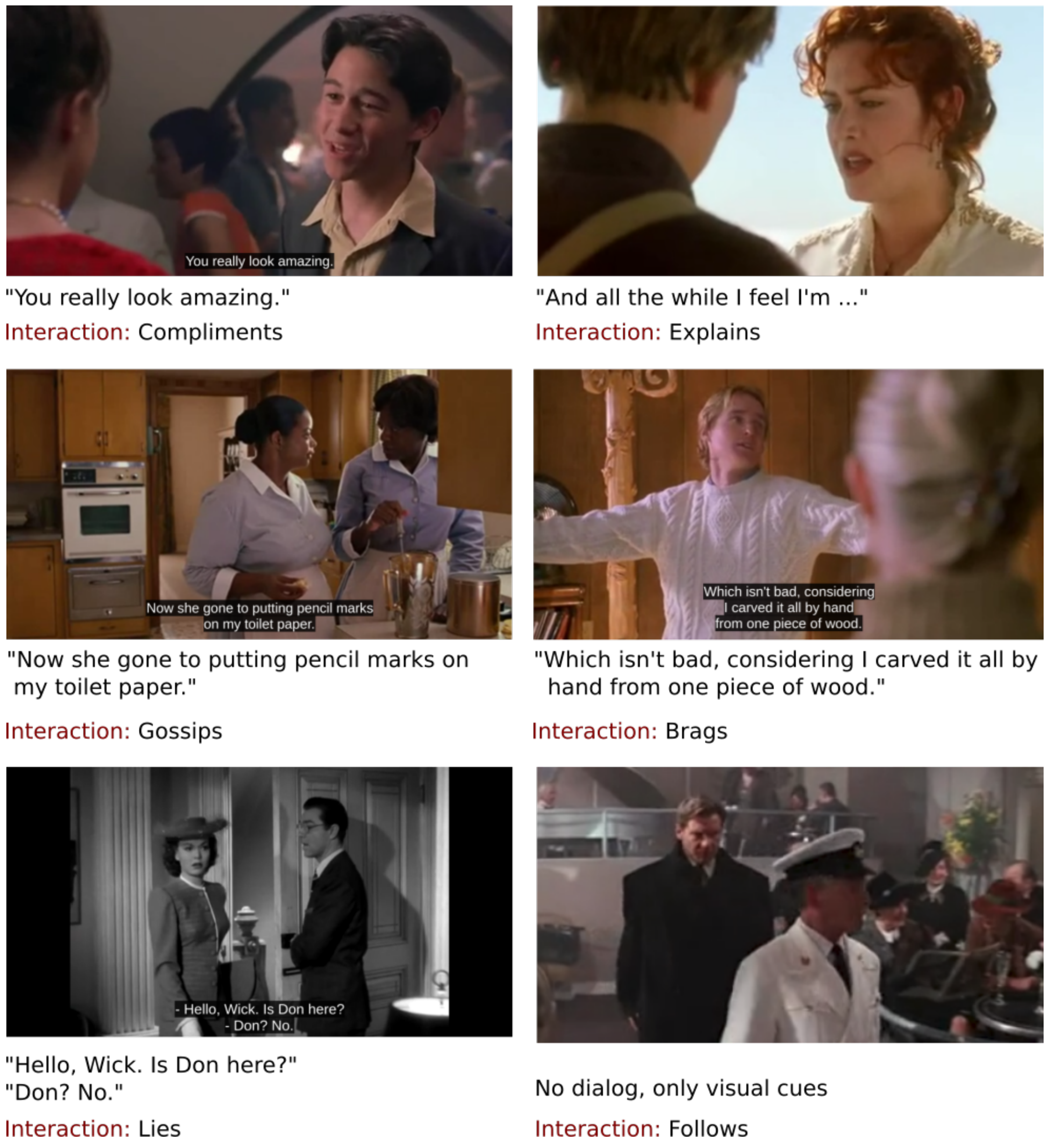


Figure 13: A tree showing possible pairwise transitions from one situation to another. For example, the situation *date* can be followed by *intimacy*. In turn, the situation *intimacy* can be followed by, among others, *talking about the past*. *Talking about the past* is followed by many kinds of situations, including *argument*. The sequence *date-intimacy-talking about the past-argument* is therefore possible, but not necessarily found in the movies we have annotated so far. This longer sequence follows from multiple pairwise transitions between situations.


You really look amazing.
Now she gone to putting pencil marks on my toilet paper.
Which isn't bad, considering I carved it all by hand from one piece of wood.
- Hello, Wick. Is Don here? - Don? No.

"You really look amazing."
Interaction: Compliments

"And all the while I feel I'm ..."
Interaction: Explains

"Now she gone to putting pencil marks on my toilet paper."
Interaction: Gossips

"Which isn't bad, considering I carved it all by hand from one piece of wood."
Interaction: Brags

"Hello, Wick. Is Don here?"
"Don? No."
Interaction: Lies

No dialog, only visual cues
Interaction: Follows

Figure 14: Examples of interaction annotations. All interactions but the bottom right are inferred mainly from dialog (shown under each image); the bottom right interaction is based only on visual cues.

References

[1] Feeling Vocabulary, http://www.ryerson.ca/~jgingras/pdf/Feeling%20Vocabulary.pdf.

[2] T. Baltrušaitis, P. Robinson, and L.-P. Morency. OpenFace: An Open Source Facial Behavior Analysis Toolkit. In *Applications of Computer Vision (WACV)*, 2016.

[3] 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.

[4] P. Ekman. An Argument for Basic Emotions. *Cognition & Emotion*, 6(3-4):169–200, 1992.

[5] M. Guillaumin, J. Verbeek, and C. Schmid. Is that You? Metric Learning Approaches for Face Identification. In *ICCV*, 2009.

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

[7] F. Schroff, D. Kalenichenko, and J. Philbin. FaceNet: A Unified Embedding for Face Recognition and Clustering. In *CVPR*, 2015.

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

[9] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *CVPR*, 2001.