Supplementary Material: Towards Long-Form Video Understanding

Chao-Yuan Wu   Philipp Krähenbühl

The University of Texas at Austin

A. Supplementary Implementation Details

Object Transformer Architecture. We develop our default architecture based on the BERT$_{\text{BASE}}$ architecture from Delvin *et al*. [2]. Specifically, our architecture uses 12 attention heads per layer in a 3-layer[1] design, with 64-dimensional attention heads, 768-dimensional hidden layers, and 3072-dimensional feed-forward networks. Since each example contains a different number of instances of different lengths, we perform attention masking as typically implemented in standard frameworks [12]. Each example contains up to 256 tokens when using person features only (default) and up to 512 tokens when additionally using object features. To construct positional embeddings, we first encode the position into a three-value vector, (distance from start, distance from end, distance from center), and use a linear transformation to map this vector to a 768-dimensional embedding vector. We use attention dropout of 0.1 following standard practice [2].

End-Task Fine-Tuning Details. Similar to prior work in natural language processing [2,8], we select the batch size $\in \{16, 32\}$ and the number of training epochs from {3, 5, 10, 20, 30, 50, 100, 200, 300, 500, 1000, 2000} for each task on the validation set. The selected hyperparameter (epochs; batch size) is (20; 16) for '*relationship*', (20; 32) for '*way of speaking*', (300; 16) for '*scene/place*', (500; 32) for '*likes*', (300; 16) for '*views*', (500; 32) for '*director*', (5; 16) for '*genre*', (500; 16) for '*writer*', and (1000; 16) for '*year*'.

'R101-SlowFast+NL' Baseline Implementation Details. We use the open-source PySlowFast codebase [4] for this 3D CNN baseline. The Kinetics-600 [1] and AVA [6] pre-training model weights are from the PySlowFast Model Zoo[2]. We observe that 3D CNNs are more sensitive to learning rates than transformers. We thus additionally select learning rate $\in \{0.001, 0.010, 0.025\}$ for each task. Linear warmup is used for the first 5% of the training schedule followed by a cosine learning rate decay following standard practice [5].

VideoBERT Implementation Details. The main difference between Object Transformers and VideoBERT [10] lies in the object-centric *vs*. frame-centric view in design. In our experiments, we aim at comparing this central design choice, while controlling minor implementation details. To this end, we use the same positional embedding formulation as ours for VideoBERT for fair comparison. In addition, Sun *et al*. [9] notes that continuous input vectors is advantageous over discrete tokens. We thus use dense input vectors with the same InfoNCE-style training objective for VideoBERT, instead of using discretized tokens as in their original paper, for consistent comparison to Object Transformers. We also use the same ResNet-101 [7] SlowFast [5] architecture with non-local blocks [11] as what Object Transformers use, and pre-train the model on Kinetics-600 [1]. We select the best hyperparameters for each task for VideoBERT using the same grid search protocol as Object Transformers.

B. Supplementary Dataset Details

Here we provide additional details for the 9 LVU tasks.

- **Relationship prediction** is a 4-way classification task over '*friends*', '*wife-and-husband*', '*boyfriend-and-girlfriend*', and '*ex-boyfriend-and-ex-girlfriend*'. The ground-truth labels are mined from the description associated with each video. For example, given the description, '*Rosemary (Mia Farrow) and her husband (John Cassavetes) quarrel about doctors; she feels the baby kicking.*', we can infer the '*wife-husband*' relationship for this video. This task contains 226 videos.
- **Way of speaking prediction** is a 5-way classification task over '*explain*', '*confront*', '*discuss*', '*teach*', and '*threaten*'. The labels are mined analogously to the relationship prediction task. This task contains 1,345 videos.
- **Scene/place prediction** is a 6-way classification task over '*office*', '*airport*', '*school*', '*hotel*', '*prison*', and '*restau-*

[1] Original BERT$_{\text{BASE}}$ is 12-layer, but we found 3 layers suffice for Object Transformers to achieve good performance.

[2] https://github.com/facebookresearch/SlowFast/blob/master/MODEL_ZOO.md

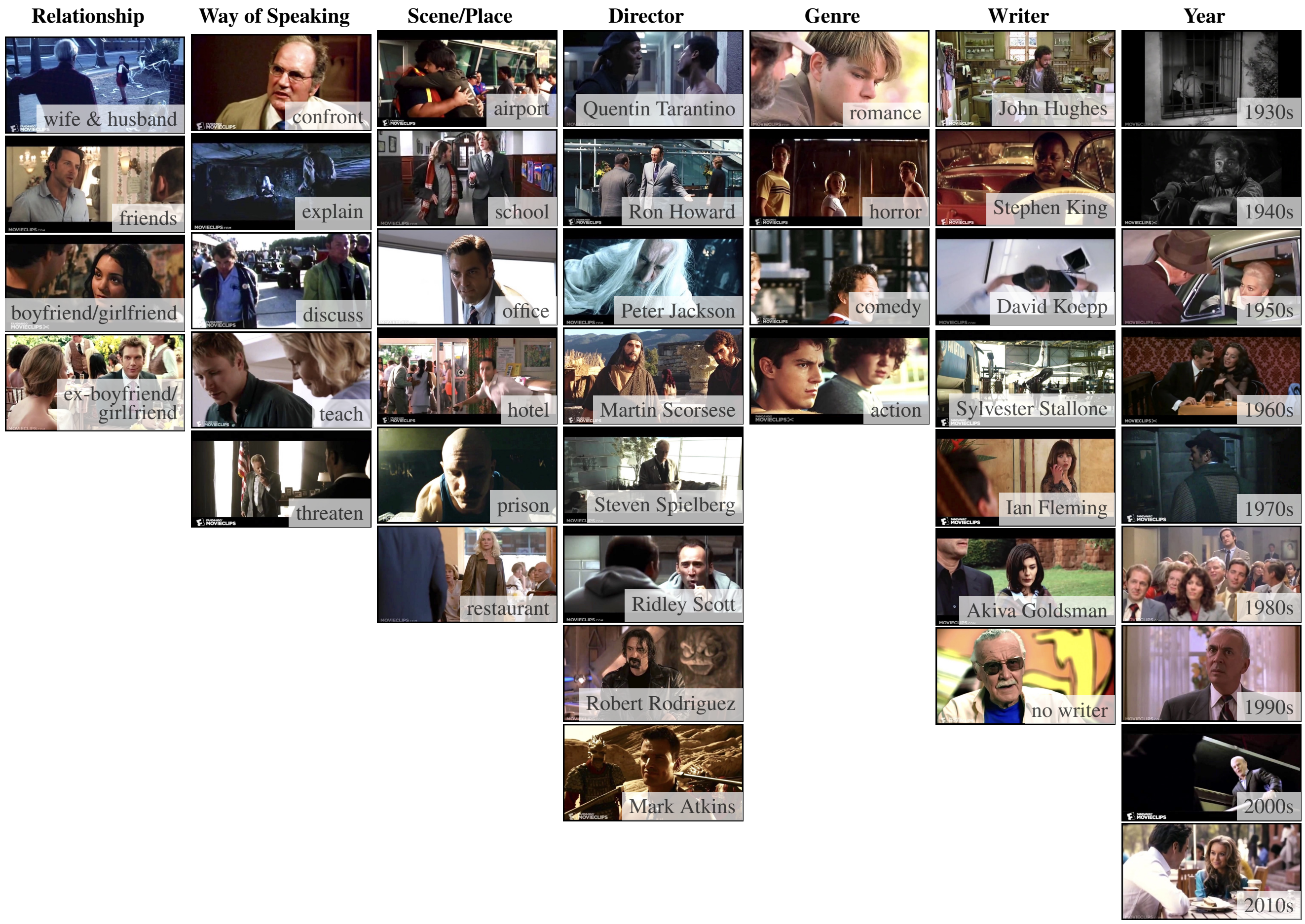

Figure A.1. **Additional Examples for Classification Tasks.** Here we present example frames for all the classes in each classification task. (Best viewed on screen.)

rant'. The labels are mined analogously to the relationship prediction task. This task contains 723 videos.

- **Director prediction** is an 8-way classification task over '*Ron Howard*', '*Martin Scorsese*', '*Steven Spielberg*', '*Quentin Tarantino*', '*Ridley Scott*', '*Peter Jackson*', '*Robert Rodriguez*', and '*Mark Atkins*'. These classes correspond to the 10 most frequent directors in our dataset, excluding Ethan Coen and Joel Coen. The Coen brothers co-direct frequently; we remove them to set up a single-label task. This task contains 950 videos.

- **Writer prediction** is 7-way classification task over '*Stephen King*', '*Sylvester Stallone*', '*John Hughes*', '*Ian Fleming*', '*Akiva Goldsman*', '*David Koepp*', and '*no writer*' (*e.g.*, documentary). They correspond to the 10 most frequent writers in our dataset, excluding Ethan Coen and Joel Coen (due the same reason we discussed above) and Richard Maibaum, whose movies largely overlap with Ian Fleming movies. This task contains 1,111 videos in total.

- **Genre prediction** is a 4-way classification task over '*Action/Crime/Adventure*', '*Thriller/Horror*', '*Romance*', and '*Comedy*'. The labels are obtained through IMDb. We exclude videos that belong to more than one of these genres. There are 4,307 videos in this task.

- **Year prediction** is a 9-way classification task over the movie release "decades", in '*1930s*', '*1940s*', . . . , '*2010s*'. The labels are obtained through IMDb. This task contains 1,078 videos.

- **YouTube like ratio prediction** is regression task to predict how much a video is "liked", namely, $\frac{likes}{likes+dislikes} \cdot 10$.[3] We access the like and dislike counts using YouTube Data API V3[4] on August 4th, 2020. We use videos with at least 30,000 votes in this task. The most liked video is the '*A Pocketful of Sunshine*' scene from movie '*Easy A*

[3] We scale the target by 10 (thus the target is in [0, 10]) for consistency with recommendation system literatures (*e.g.*, [3]), where the scale of ratings are often in [0, 10].

[4] https://developers.google.com/youtube/v3

(2010)'[5] with 32,967 likes and 242 dislikes. The least liked video is the '*Seeing an Old Friend*' scene from movie '*Enter the Ninja (1981)*'[6] with 18,595 likes and 15,432 dislikes. This task contains 940 videos in total.

- **YouTube view count prediction** is a regression task. The view counts are also accessed using YouTube API V3 on August 4th, 2020. Since the view counts follow a long-tail distribution, we predict $\log(views)$ in this task. To control the effect of video upload time, all videos in this task were uploaded to YouTube in the same year (2011). The most viewed video is the '*Kong Battles the T-Rexes*' scene from '*King Kong (2005)*'[7] with 132,862,771 views. The least viewed video is the '*Margo to the Rescue*' scene from '*The Locksmith (2010)*'[8] with 531 views. This task contains 827 videos.

For the three content tasks, we spot-checked the training set and corrected any wrong labels in the validation and test sets (∼1 in 9). Fig. A.1 presents example frames for all the classes in each classification task.

C. Qualitative Evaluation Full Sequences

In Fig. 5 of the main paper, we present subsampled frames of three examples of Masked Instance Prediction. In Fig. C.1, we present the full 60-second frames.

References

[1] Joao Carreira, Eric Noland, Andras Banki-Horvath, Chloe Hillier, and Andrew Zisserman. A short note about kinetics-600. *arXiv preprint arXiv:1808.01340*, 2018.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.

[3] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In *KDD*, 2014.

[4] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. PySlowFast. https://github.com/facebookresearch/slowfast, 2020.

[5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *ICCV*, 2019.

[6] 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 *CVPR*, 2018.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[8] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[9] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019.

[10] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *ICCV*, 2019.

[11] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *CVPR*, 2018.

[12] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's transformers: State-of-the-art natural language processing. *ArXiv preprint arXiv:1910.03771*, 2019.

[5] https://www.youtube.com/watch?v=ylvh800i85I

[6] https://www.youtube.com/watch?v=G_1jQkCRF58

[7] https://www.youtube.com/watch?v=ZYZsJYZVt5g

[8] https://www.youtube.com/watch?v=oGKr8bdx_5E

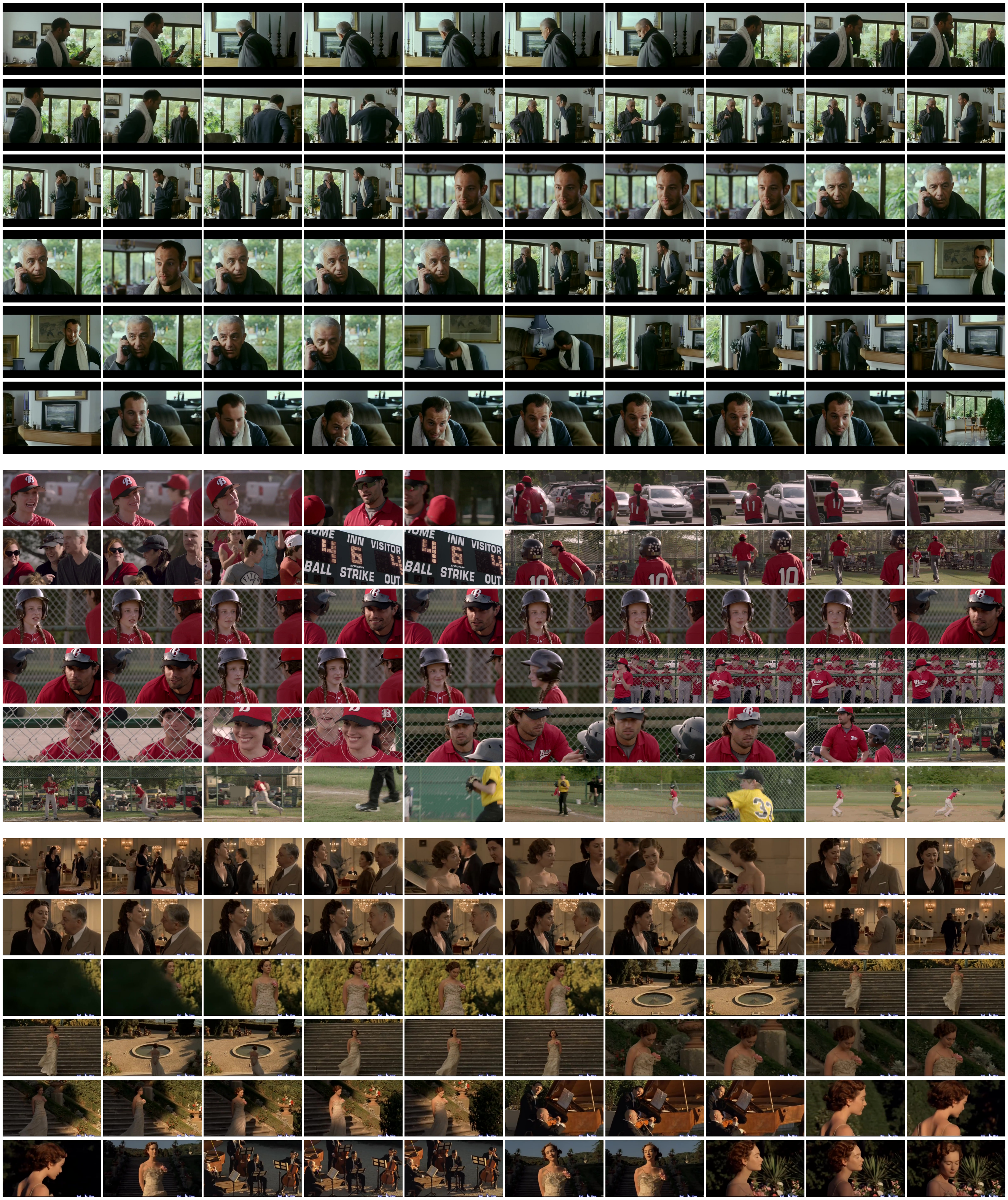

Figure C.1. **Qualitative Evaluation Full Sequences.** Here we present the full sequences for the three examples in Fig. 5 of the main paper. From left to right and then top to bottom of each group are a full 60-second sequence.