VX2TEXT: End-to-End Learning of Video-Based Text Generation From Multimodal Inputs

Appendix

Xudong Lin[1] Gedas Bertasius[2] Jue Wang[2] Shih-Fu Chang[1] Devi Parikh[2,3]
Lorenzo Torresani[2,4]
[1]Columbia University [2]Facebook AI [3]Georgia Tech [4]Dartmouth

A. Understanding the Semantics of Tokens

In order to understand the semantics of video and audio predictions learned through the end-to-end training process, we visualize the correlation between video/audio categories and generated words using WordCloud[1]. We first remove all the stop words [1] from the captions generated by VX2TEXT on the TVC test set so as to visualize the most salient words,. Then we also remove from the generated captions the words included in the corresponding input speech transcripts. By doing so, the remaining words in the generated caption are more likely to be derived from (or at least influenced by) the video/audio input. Finally, for each video/audio category, we consider all the (remaining) words generated by VX2TEXT when that category is sampled and rank them according to the TF.IDF [2] score. The TF.IDF score for each word given the sampled video/audio category is regarded as the correlation between them.

Figure 2 and Figure 3 show the visualization of 8 categories for video actions and audio events, respectively. We observe that when the categories are general enough, their literal meaning is well maintained by the training process. For example, as shown in Figure 2, kiss is one of the most correlated words to the action category "kissing" while driving and car are highly correlated to the action category "driving car." Similarly, crying is one of the most correlated words to the audio event category "crying, sobbing."

For category names that are not immediately useful for captioning, we observe interesting semantic shifts of the category names as a result of the end-to-end learning. For example, as shown in Figure 2, kitchen and table are the most correlated words to the action category "tossing salad" while couch and bed are heavily correlated to the action category "situp." Similarly, as seen in Figure 3, apartment is highly correlated to the audio event category "ding" while police is correlated to the audio event category "explosion."

These visualizations suggest that VX2TEXT does learn informative embeddings for video and audio categories and successfully integrates these embeddings for open-ended text generation.

B. Mapping from each modality to the full vocabulary

In this section, we provide more details on how to implement VX2TEXT without constraining each modality to use a small predefined vocabulary. For example, for the case of video, instead of using the original final fully-connected layer of the video backbone which projects the 512-D features into the Kinetics vocabulary of 400 action classes, we replace it with a new fully-connected layer that maps the 512-D features into the full vocabulary used by T5 which consists of 32128 tokens. In this way, our VX2TEXT is not only able to predict tokens within the predefined video vocabulary but possibly other related words in the much larger text vocabulary.

We pretrain this new fully-connected layer on TVQA by forcing it to predict the words of predicted K_v categories, with a learning rate of 0.001. The Audio branch is implemented in the same way. On TVQA, we observe that this implementation achieves comparable results (74.0%) to those obtained with the default method presented in the main paper (74.9%). We believe pretraining on larger-scale datasets will further boost the performance of this more general scheme. We leave this for future work.

C. Training, validation and test setups

On AVSD, we follow the common practice of training on the training split, using the validation set for ablation studies, and we report performance on the test set for comparison with the state-of-the-art. On TVQA, we train our model on the training split, use the validation set for ablation studies and we compare results on both the validation and the test set with the state-of-the-art. We adopt top-1 accuracy as the standard evaluation metric. On TVC, the training split is

[1] https://github.com/amueller/word_cloud

used for training and the performance is evaluated on both the validation and the test set.

D. Hyper-Parameter Tuning: K_v and K_a

In Figure 1 of this Appendix we show the TVQA validation results obtained by varying the number of video categories (K_v) and the number of audio categories (K_a) provided as input to the encoder. In order to reduce the computational cost of this experiment, we adopt a two-stage tuning strategy. We first exclude audio from the input to VX2TEXT and only explore the best values for K_v. Based on these results, we fix $K_v = 12$ and subsequently search for the optimal K_a by adding audio to the input. It can be seen that $K_a = 6$ produces the best performance. We leave the joint optimization of both hyper-parameters for future work. We also leave the hyper-parameter tuning on the embedding dimensions of video/audio categories for future work. Since the video and audio category names contain at most 13 and 15 text tokens, respectively, we simply set the dimensionality of the video embedding to be 13×768 and that of the audio embedding to be 15×768, We pad short category names with text padding tokens.

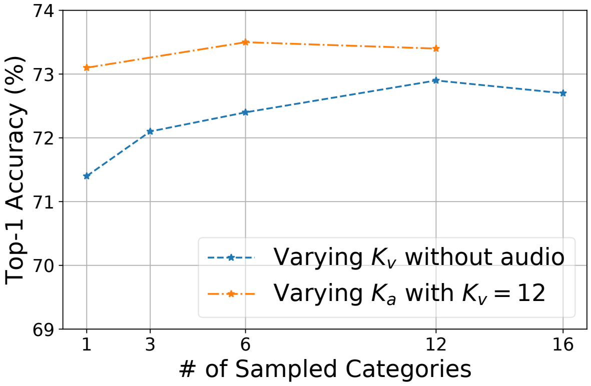


Figure 1. Impact of different number of top video categories (K_v) and top audio categories (K_a) on the performance of VX2TEXT for the task of video question answering using the validation set of TVQA.

E. Details about Generative Training Variants

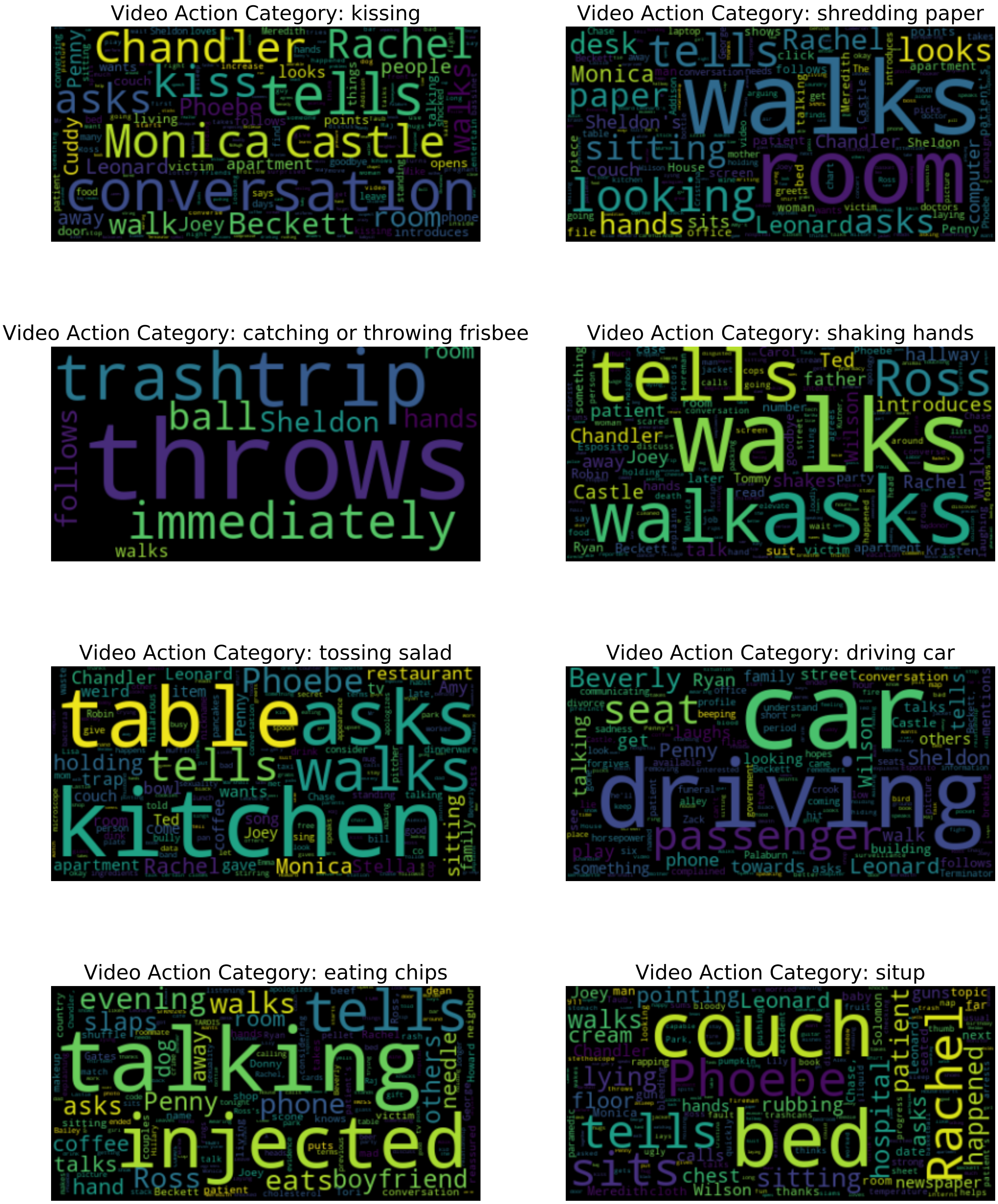
For "Generative (Question Answering & Generation)" and Cycle-consistency, we simply average (with equal weights) all the loss terms in the final training objective. The losses for Question Generation, Answer-Consistency, and Question Consistency are all following standard cross-entropy loss, as described in Equation 6 of the main paper. For Cycle-consistency, we use greedy search to decode question or answer sentences during training. Note that for simplicity, we do not apply the Gating Mechanism and Late Activation as described in [3] but instead we use the generated questions from the first epoch for Cycle-consistency training.

References

[1] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.", 2009. 1

[2] Anand Rajaraman and Jeffrey David Ullman. *Mining of massive datasets.* Cambridge University Press, 2011. 1

[3] Meet Shah, Xinlei Chen, Marcus Rohrbach, and Devi Parikh. Cycle-consistency for robust visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6649–6658, 2019. 2


Video Action Category: kissing

Video Action Category: shredding paper

Video Action Category: catching or throwing frisbee

Video Action Category: shaking hands

Video Action Category: tossing salad

Video Action Category: driving car

Video Action Category: eating chips

Video Action Category: situp

Figure 2. Cloud visualizations of words generated by Vx2Text on the TVC test set conditioned on a specific top action category sampled from the video (printed in the title above the word cloud). The higher the correlation, the larger the font.

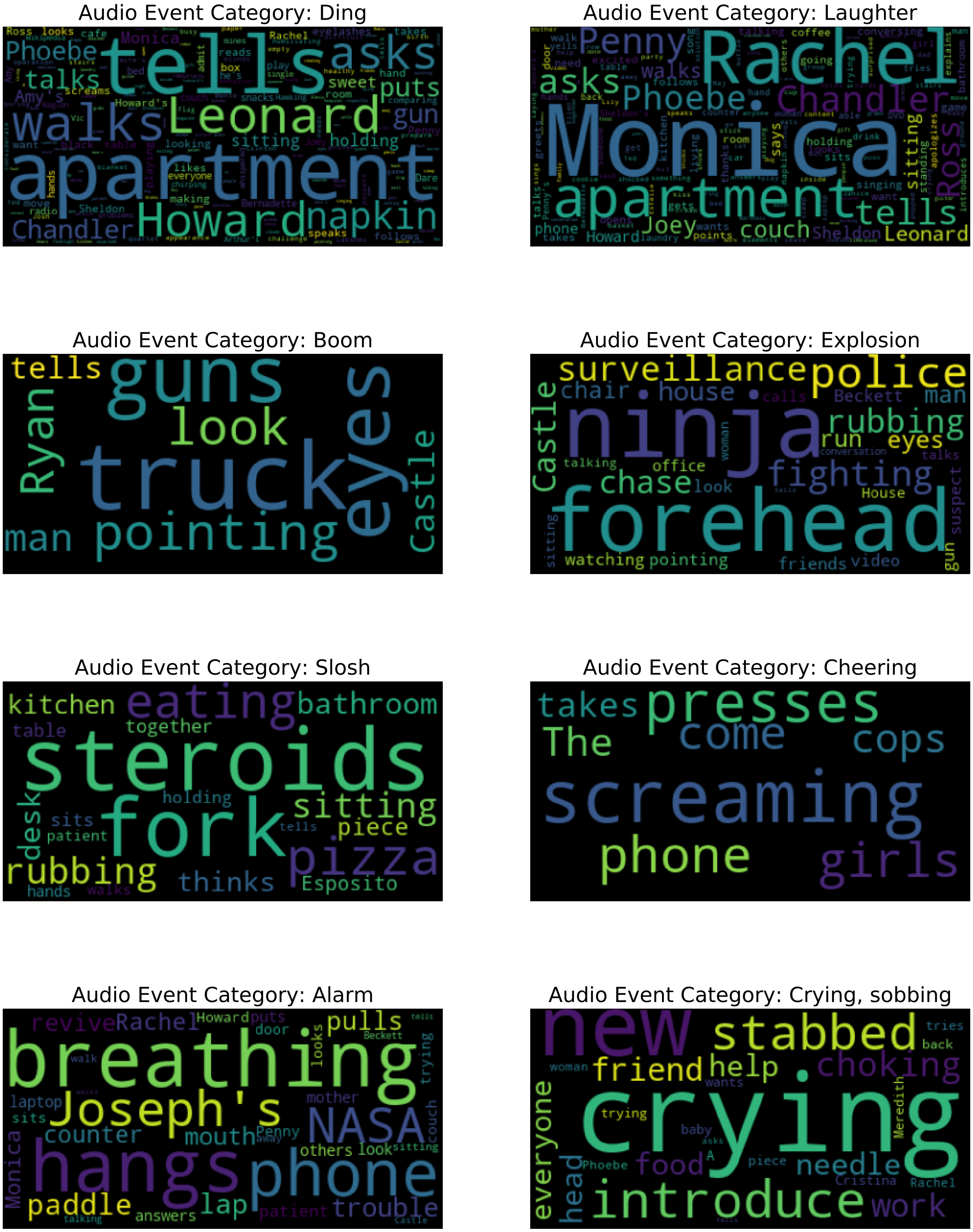


Figure 3. Cloud visualizations of words generated by VX2TEXT on the TVC test set conditioned on a specific top sound category sampled from the audio (printed in the title above the word cloud). The higher the correlation, the larger the font.