Exploring the Role of Audio in Video Captioning Supplementary Materials

Yuhan Shen[†] * Linjie Yang[‡] Longyin Wen[‡] Haichao Yu[‡] Ehsan Elhamifar[†] Heng Wang[‡]
[†] Northeastern University [‡] ByteDance

In this supplementary material, we include:

- Additional implementation details.
- Details on audio relevance analysis.
- Additional qualitative results.

1. Additional Implementation Details

Paragraph Video Captioning on ActivityNet-Captions: ActivityNet-Captions contains one reference for each video in the training set and two references for each video in the validation/testing set. Following [12], we use ground-truth segments and sentences for training. For validation/testing, we use the ground-truth segments from the first reference in validation/testing set and evaluate against the two references.

Video Encoder: We sample 16 frames from each video clip. We equally divide each video clip into 16 segments and randomly sample 1 frame from each segment during training. For testing, we uniformly sample 16 frames. The frames are resized and cropped into images of size 224×224. Each video clip with the size of $16\times224\times224\times3$ is fed into the Video Swin Transformer [8] initialized with the weights pre-trained on Kinetics 600 [1] and tokenized into $N_v = 8 \times 7 \times 7 = 392$ video tokens. Following prior works [7, 8], given the raw video frames of the size $T \times H \times W \times 3$, where T is the number of frames, $H \times W$ is the image height and width, and 3 is the RGB channels, the size of the output features of Video Swin Transformer is $\frac{T}{2} \times \frac{H}{32} \times \frac{W}{32} \times 8C$, where C is the channel dimension. In our experiments, the input size of the video encoder is $16 \times 224 \times 224 \times 3$ and the channel dimension is $C = 128$, so the output size is $8 \times 7 \times 7 \times 1024$. To be consistent with the token dimension of the other modules, we add a linear layer to map the feature dimension into 768. Hence, the number of video tokens is $N_v = 8 \times 7 \times 7 = 392$ and the token dimension is $D = 768$.

Audio Encoder: Our implementation of the audio encoder is similar to that in [11]. Each audio is resampled to 22,050Hz and divided into frames of 1536 samples with hop length of 588. Then we apply 64 mel-scale filters and take logarithm on the amplitude to get the log mel spectrogram. To tackle the variable length of audios, we set a maximum length of frames as 256 ($\sim$6.8 sec.). The audio clips shorter than 256 frames will be zero-padded and longer clips will be down-sampled to be 256 frames. As a result, the dimension of input mel-spectrogram is 256×64 for each audio. We use a 12-layer Transformer with 12 attention heads on audio spectrogram. We apply the linear 1-dimensional layout on the spectrogram as in [11] rather than a two-dimensional (image-like) one in [4], as we notice 1-dimensional layout is more suitable for fine-grained audio events like speech. The Transformer produces audio features of dimension 256×768 and then, following [11], we apply an average pooling by a factor of four to resize the sequence to a length of $N_a = 64$ audio tokens. During training, we use time and frequency masking as in SpecAugment [10] for data augmentation. The time mask parameter is set as 32 and the frequency mask parameter is set as 128.

Cross Encoder: We use a 3-layer Transformer with 12 attention heads as the cross-modal encoder. The feature embeddings of different modalities will be added with the position embedding and token type embedding to distinguish the position and modality of the tokens.

Training Details: We pre-train the model on HowTo100M for 100 epochs using Adam optimizer [6]. For each video, we sample three video-caption pairs from a long video in one iteration. Empirically, we found the MBP loss not sensitive to the two hyperparameters β and α. We set $\beta = 0.99$ and $\alpha = 10$ for all the experiments. We pre-train the model on 64 Nvidia Tesla A100 GPUs and it takes around five days. The base learning rate is 10^{-4} and we use a linear decay learning rate schedule with a warm-up of 10% training epochs as in [9]. For fine-tuning, we set the initial learning rate as 10^{-5}. We train the model for 5 epochs on YouCook2 and MSRVTT, 10 epochs on VATEX, and 30 epochs on ActivityNet-Captions.

*Work done during YS's internship at ByteDance.

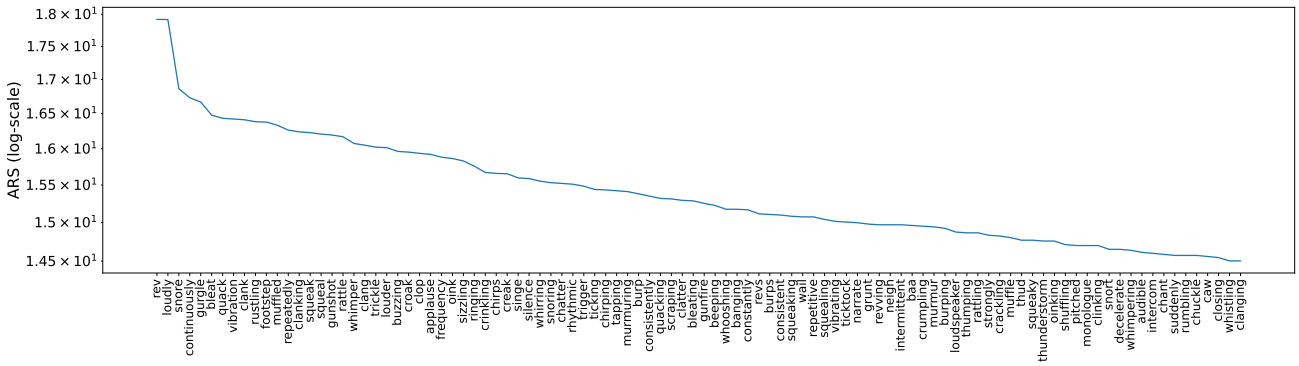

Figure 1. The ARS of top 100 words.

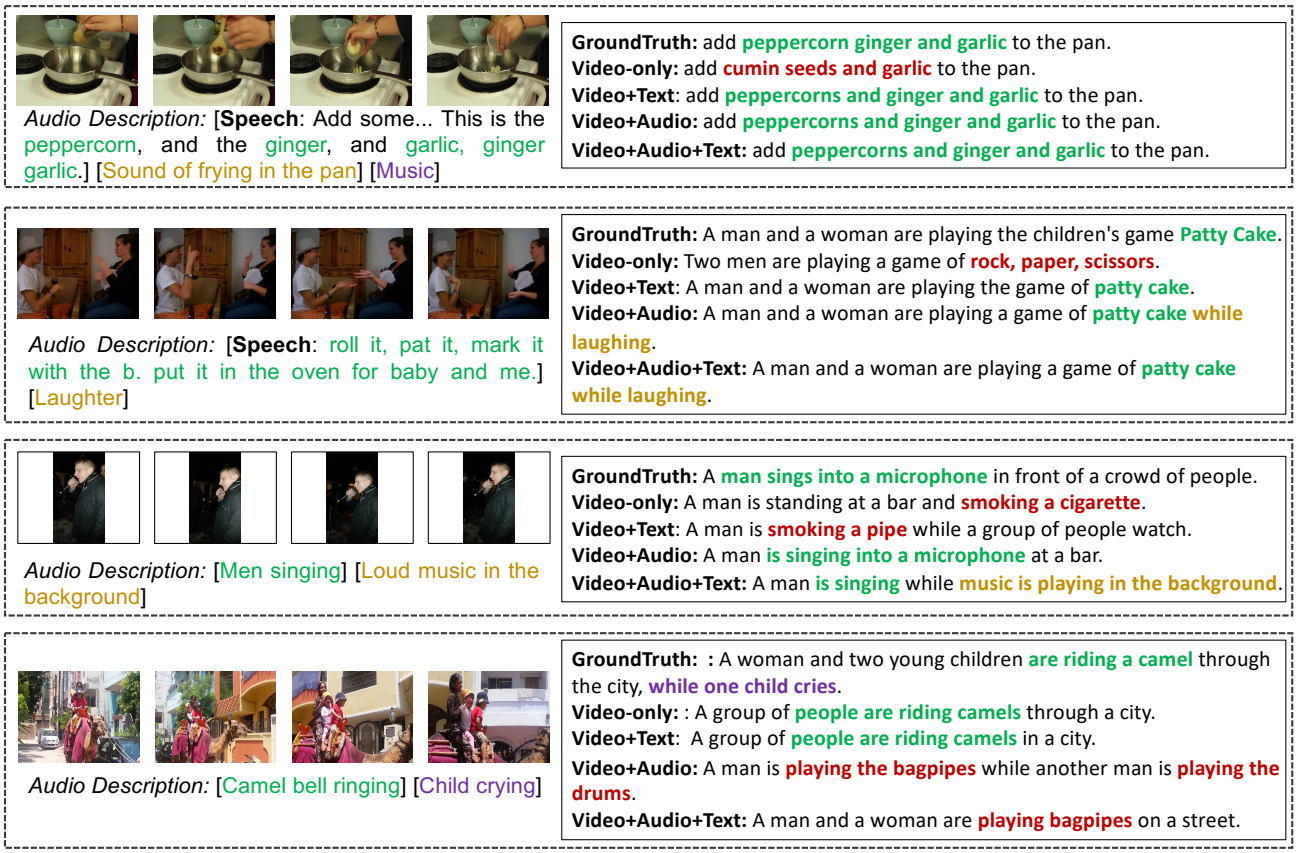

Figure 2. Some qualitative examples when we input different modalities to generate captions.

2. Audio Relevance Analysis

We design two metrics, *i.e*. Speech Coverage Rate (SCR) and Audio Relevance Score (ARS), to measure how relevant the captions are to speech and audio.

For SCR, we tokenize the annotated captions and speech transcripts, and compute the percentage of tokens in captions that are covered by the associated speech transcripts.

For ARS, we collect the captions from audio captioning and image captioning datasets. We calculate the word frequencies in audio captioning and image captioning datasets, and assign a higher score to the words that occur more frequently in audio captions than image captions. We use the annotated captions from AudioCaps [5] and Clotho [3] as audio captions and the annotated captions from CoCo-Captions [2] as image captions. We tokenize and lemmatize

	YouCook2	MSRVTT	VATEX	ActivityNet
SCA	**48.67**	14.7	5.53	12.63
ARS	1.202	2.903	**3.692**	3.487

Table 1. The Speech Coverage Rate (SCA, %) and Audio Relevance Score (ARS) on the downstream datasets.

all captions, and remove all punctuation and stop words. Then we compute the word frequency in audio captions and image captions. Let $f^a(\mathbf{w})$ and $f^i(\mathbf{w})$ denote the frequency of word $\mathbf{w}$ in audio captions and image captions respectively, then ARS is computed as:

$$\text{ARS}(\mathbf{w}) = \max(\log \frac{f^a(\mathbf{w})}{f^i(\mathbf{w})}, 0). \quad (1)$$

Hence, a higher ARS will be assigned to those words whose frequency is much higher in audio captions than that in image captions. On the contrary, if a word is more frequent in image captions, then the log of the division is negative, and the ARS will be set as zero. The ARS of a sentence is the sum of the ARS of all non-stop words in it, and we compute the average ARS of all captions on each dataset.

Figure 1 shows the ARS of the top 100 words. We can see that most of those words are highly relevant to audio modality, including verbs that describe a certain type of sound, *e.g. "snore", "gurgle", "bleat"*, adjectives or adverbs that describe the pattern of sound, *e.g. "loudly", "muffled"*, or nouns that are usually associated with a type of sound, *e.g. "vibration", "gunshot"*, *etc*.

Tab. 1 lists the SCR and ARS on downstream datasets. We notice that a large portion of the captions on YouCook2 are mentioned in speech. Conversely, though only a small amount of captions are covered by speech, the captions on VATEX are most relevant to audio modality.

3. Additional Qualitative Results

Qualitative results. We show some qualitative examples in Fig. 2 by comparing the ground truth, and predictions of our model with video-only, video+text, video+audio, and video+audio+text inputs. In the first example, when only visual modality is available, the model misclassifies the peppercorns as cumin seeds as they are hardly visible in the video. By adding audio and/or ASR text as input, the model correctly generates the caption because the ingredients are clearly introduced in the speech, which shows that we can infer the ingredient from raw audio without relying on an off-the-shelf ASR system. In the second example, the caption is not explicitly described in the audio, but the model is able to correct its prediction from "rock, paper, scissors" to "patty cake" by adding audio or text modality as it can infer the type of game from the speech. In particular, when we add audio as the input, the model not only predicts the correct type of game, but also provides more information about the audio, *i.e*. recognizing that the people are laughing. The third example shows that audio is also helpful when there is no speech in the video. Using video-only or video+text as the inputs, the model generates incorrect captions, *e.g*. "smoking a cigarette" or "smoking a pipe" as the microphone is not quite visible. However, if we add audio as the input, the model will correctly detect that the man is singing and music is playing in the background. The last example is a failure case, where the audio-visual model mistakenly recognizes the sounds of camel bell ringing and child crying as the sounds of the bagpipes and the drums. It indicates that the model gives too much attention to the audio modality and ignores the visual appearance of people riding a camel.

Figure 3. Additional visualization on the attention maps from audio modality to the RGB space for global cross fusion and local fusion modules on YouCook2, MSRVTT, VATEX, and ActivityNet, respectively.

Attention maps. We show additional visualization on the attention maps on the four downstream datasets respectively. Similar to what we observed in the main paper, in the first and the fourth example, where the items in the videos are mentioned in the speech, *i.e*., "*bacon*" and "*hair*", the

local attention will focus on the regions of those items. In the second example, where there is sound of singing in the video, both local and global modules will give attention to the performers on the stage, and we notice that the attention of the global module will be more concentrated. In the third example, where there is sound of speaking and laughter, the global module focuses more on the mouth region, where the sound comes from.

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. 1

[2] Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015. 2

[3] Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE, 2020. 2

[4] Yuan Gong, Yu-An Chung, and James Glass. AST: Audio Spectrogram Transformer. In *Proc. Interspeech 2021*, pages 571–575, 2021. 1

[5] Chris Dongjoo Kim, Byeongchang Kim, Hyunmin Lee, and Gunhee Kim. Audiocaps: Generating captions for audios in the wild. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 119–132, 2019. 2

[6] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 2015. 1

[7] Kevin Lin, Linjie Li, Chung-Ching Lin, Faisal Ahmed, Zhe Gan, Zicheng Liu, Yumao Lu, and Lijuan Wang. Swinbert: End-to-end transformers with sparse attention for video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17949–17958, 2022. 1

[8] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3202–3211, 2022. 1

[9] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Jason Li, Taroon Bharti, and Ming Zhou. Univl: A unified video and language pre-training model for multimodal understanding and generation. *arXiv preprint arXiv:2002.06353*, 2020. 1

[10] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. Specaugment: A simple data augmentation method for automatic speech recognition. *Proc. Interspeech 2019*, pages 2613–2617, 2019. 1

[11] Rowan Zellers, Jiasen Lu, Ximing Lu, Youngjae Yu, Yanpeng Zhao, Mohammadreza Salehi, Aditya Kusupati, Jack Hessel, Ali Farhadi, and Yejin Choi. Merlot reserve: Neural script knowledge through vision and language and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16375–16387, 2022. 1

[12] Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. End-to-end dense video captioning with masked transformer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8739–8748, 2018. 1