# RUC_AIM3 at TRECVID 2021: Video to Text Description

**Liang Zhang, Yuqing Song and Qin Jin***
School of Information, Renmin University of China
{zhangliang00, syuqing, qjin}@ruc.edu.cn

## Abstract

In this report, we present our solutions for the Video to Text Description (VTT) task in TRECVID 2021 [1], which includes two sub-tasks, Description Generation and Fill-in-the-Blanks. For the Description Generation sub-task, we employ a Concept Enhanced Pretraining-based Transformer Model (CE-PTM), which is trained via the Video-guided Masked Language Modeling (VMLM) objective. Experimental results show that with the proposed pre-training method, the transformer model outperforms LSTM-based models for short video description generation. For the Fill-in-the-Blanks sub-task, we automatically build the training and validation set by randomly blanking nouns and verb phrases of the video captions. We further propose 4 generation approaches based on the same pre-trained PTM to generate phrases to fill in the blank. Our team RUC_AIM3 finally ranks 1st place in Description Generation sub-task on METEOR, CIDEr, SPICE, and STS metrics, and the 1st place in Fill-in-the-Blanks sub-task by Human Evaluation in TRECVID 2021.

## 1 Description Generation

### 1.1 Approach

The description generation subtask aims to generate a natural language sentence to describe the content given a short video [2]. Previous works suggest that the two-layer LSTM model is more feasible than the vanilla transformer framework for generating short-video descriptions [2, 3]. However, pre-training based transformer models, such as VideoBERT [4], UniVL [5], Oscar [6] and CLIP [7] have achieved great success in various vision-language tasks in recent years. Therefore, we wonder whether the pre-training based transformer model is also suitable for short video captioning? To answer this question, we design a Concept Enhanced Pretraining-based Transformer Model (CE-PTM) trained with the Video-guided Masked Language Modeling (VMLM) task. As shown in Figure 1, CE-PTM consists of four parts: Video Encoder, Text Encoder, Concept Encoder, and Multimodal Transformer.

**Video Encoder** is employed to encode raw video frames into visual representations. It consists of three parts: an off-the-shelf feature extractor, linear projection layer, and a transformer encoder layer. Considering the huge amounts of spatio-temporal information contained in a video, training the model in an end-to-end manner can consume a lot of computing resources. Therefore, we sample key-frames from videos and extract video features with off-the-shelf networks. We sample one key frame every 8 frames. In order to comprehensively encode videos, we extract five types of video features with different backbone networks, including I3D [8], ResNeXt-101 [9], irCSN [10], Swin-Transformers [11], and CLIP ViT-B/32 [7]. The comparisons between these features are shown in Table 1. For each sampled frame, we concatenate the extracted five feature vectors and finally get

*Qin Jin is the corresponding author.

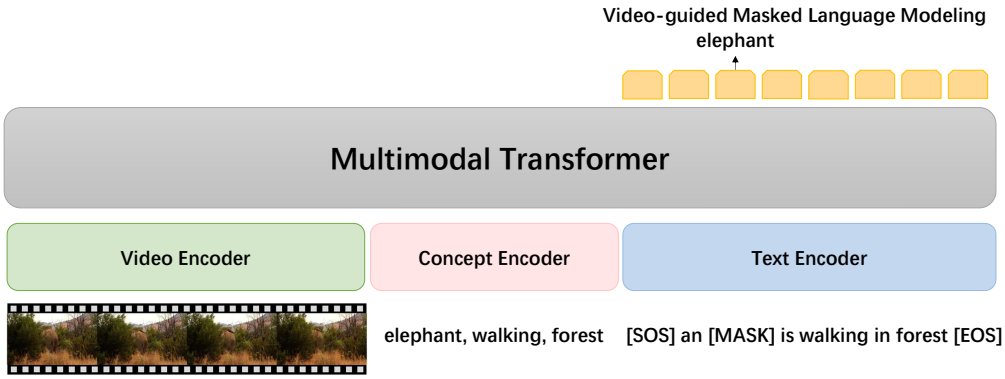


Figure 1: Architecture of CE-PTM.

a 7168-dimensional[2] video feature per key-frame. Therefore, the video is represented as a sequence of $V^f = \{v_1^f, \ldots, v_{L_v}^f\} \in \mathbb{R}^{L_v \times 7168}$, where $L_v$ denotes the number of key frames of the video. A linear layer is then employed to project the video features into 512-dimensional embedding vectors. To distinguish different modalities, we further add type embedding to each visual appearance feature. Then, we encode the video embeddings into visual representations $V^r$ with a one-layer transformer encoder. Note that we do not add the positional embedding since our video embeddings already contain 3D features with temporal information [3]. Formally, the visual representations are calculated as follows:

$$V^e = V^f W_e^T \tag{1}$$

$$V^r = \text{TransformerEncoderLayer}(V^e + \text{TE}(0), \theta_v) \tag{2}$$

where TE$(\cdot)$ represents Type Embedding layer and $\theta_v$ represents parameters of the TransformerEncoderLayer. $W$ represents a linear projection layer if not mentioned specially.

Table 1: Comparison between the video features.

| Name | Type | Architecture | Pretrained Data | Dimension |
|---|---|---|---|---|
| I3D [8] | 3D | CNN | Kinetics-400 [8] | 1024 |
| ResNeXt-101 [9] | 2D | CNN | ImageNet [12] | 2048 |
| irCSN [10] | 3D | CNN | IG-65M [13] | 2048 |
| Swin-Transformers [11] | 2D | Transformer | ImageNet [12] | 1536 |
| CLIP ViT-B/32 [7] | 2D | Transformer | WebImageText [7] | 512 |

**Text Encoder** is proposed to encode sentences into textual representations. We use a learnable embedding matrix to generate word embeddings $T^e = \{t_1^e, ..., t_{L_t}^e\}, T^e \in \mathbb{R}^{L_t \times 512}$. Then, we add positional embeddings and type embeddings to $T^e$. Similar with the video encoder, we encode text embeddings with a one-layer transformer encoder and get the final textual representations $T^r$, which is calculated as follows:

$$T^r = \text{TransformerEncoderLayer}(T^e + \text{TE}(1) + \text{PE}, \theta_t) \tag{3}$$

where PE denotes Positional Embedding as in [14], and $\theta_t$ represents the parameters of TransformerEncoderLayer$(\cdot)$.

**Concept Encoder** has the same architecture as the Text Encoder, and shares the same word embedding layer. We name the objects appearing in the video and the actions performed in the video as video concepts, which deliver important information of the video content. We first automatically generate video-concept pairs based on the video captioning dataset, by extracting nouns and verbs from captions. Then, we train an LSTM model to predict concepts based on the video content. After

[2] 7168=1024+2048+2048+1536+512

[3] We find adding the positional embedding performs slightly worse in our experiments.

training, we employ the LSTM model as an off-the-shelf concept extractor to predict concepts for the video and treat them as an additional input to the model. The predicted concepts are then fed into the concept encoder to get conceptual representations denoted as $C^r$.

**Multimodal Transformer** is proposed to encode representations in different modalities and explore the interaction between each other. It consists of four transformer encoding layers. Formally, we concatenate visual, conceptual and textual representations as the input sequence of Multimodal Transformer as follows:

$$H_0 = [V^r; C^r; T^r] \tag{4}$$

$$H_i = \text{TransformerEncoderLayer}(H_{i-1}, \theta_i), 0 < i <= 4 \tag{5}$$

Suppose the output of the last layer is denoted as $H_4 = [h_1^v, ..., h_{L_v}^v, h_1^c, ...h_{L_c}^c, h_1^t, ...h_{L_t}^t]$, $H_4 \in \mathbb{R}^{(L_v+L_c+L_t)\times 512}$, we only use the textual hidden state $H_t = [h_1^t, ...h_{L_t}^t] \in \mathbb{R}^{L_t \times 512}$ in the following Video-guided Masked Language Modeling task.

**Pretraining Task.** The task we use in the pretraining stage is called Video-guided Masked Language Modeling (VMLM). As shown in Figure 1, VMLM aims to predict the masked caption words based on both the video content and sentence context. We randomly choose 15% of words for prediction. If a word is chosen, we replace it with (1) the special word [MASK] 80% of the time, (2) a random token in the vocabulary 10% of the time, and (3) keeping unchanged 10% of the time following BERT [15]. The hidden state from the Multimodal Transformer is fed to an MLP layer to predict the original word. We train the VMLM task with the cross-entropy loss as follows:

$$L_{vmlm} = \frac{1}{N} \sum_{t_i^m = [\text{MASK}]} \log p(t_i | V, C, T_{\setminus i}^m) \tag{6}$$

where $T = [t_1, ..., t_{L_t}]$ refers to the original sequence. $T^m = [t_1^m, ..., t_{L_t}^m]$ refers to the masked sequence, and $N$ refer to the number of masked words.

**Finetuning and Inference.** Different from the architecture in [16], which uses a decoder to generate the sequence, we generate the caption based on the same transformer encoder following the strategy in [17]. We finetune the model with a variant of the VMLM task, which constrains the self-attention mask to avoid the model seeing the future words. During inference, we first encode the video, concepts, and special start token [SOS] as input. Then a [MASK] token is input and the model starts to predict the first word. Once a word is predicted, the [MASK] token is replaced with the predicted word, and a new [MASK] token is input to model for predicting the next word. We generate the caption in this manner auto-regressively until the special end token [EOS] is predicted.

We further improve the model by reinforcement learning (RL) with CIDEr [18] reward to address the exposure bias and target mismatch problems [19]. The RL Loss is defined as:

$$L_{rl} = -\frac{1}{L_s} r(T^s) \sum_{i=1}^{L_s} \log p(T_i^s | V, C, T_{<i}^s) \tag{7}$$

where $T^s$ is the sequence generated by the model during training using the inference strategy described above, and $r(\cdot)$ denotes CIDEr [18] score.

**Hybrid Reranking.** Considering that the captions generated by different models can be complementary, we train three types of models in total: LSTM-based BUTD, PTM, and CE-PTM. We late fuse the outputs of these models with a hybrid reranking. We train an off-the-shelf video-text matching model based on VSE++ [20] to evaluate the visual relevance of the generated captions. We treat the visual relevance matching scores between the generated captions and videos as the confidence scores of our system. Then we rerank the captions generated from the three types of models by the confident scores and choose the best description for each video.

### 1.2 Experiments

We employ the TGIF [21], MSRVTT [22], VATEX [23] and TRECVID VTT 2016-2019 video captioning datasets as the training set, and TRECVID VTT 2020 as the validation set. To verify the effectiveness of the proposed model, we conduct experiments to compare performances between TOP1-in-2020 [3], BUTD [24], our Pretraining-based Transformer Model (PTM), and Concept

Enhanced Pretraining-based Transformer Model (CE-PTM). Note that we denote the model without Concept Encoder as PTM, which does not use concepts as input.

Table 2 shows the captioning results of different models. Note that TOP1-in-2020 uses the same architecture of BUTD, but it encodes videos with ResNext-101 and irCSN only [3]. The comparison between TOP1-in-2020 and BUTD shows that adding different kinds of features brings improvement. Our PTM model is shown to achieve significantly higher scores on all the metrics compared with BUTD. Although the work in [3] shows that vanilla transformer architecture is inferior to LSTM models when generating short video descriptions, we demonstrate that with our pre-training strategy, the transformer model could perform better. Comparison between PTM and CE-PTM shows that introducing concepts into the model brings limited improvement to the single model performance. However, adding the captions generated by CE-PTM in the hybrid reranking stage could bring more performance gains, which demonstrates that the automatically predicted concepts help the model to generate more diverse captions and they are complementary to the classic captioning models.

Table 2: Models performances on TRECVID VTT 2020 dataset. Note that PTM and CE-PTM is pre-trained from-scratch using the same dataset with BUTD.

| Models | BLEU@4 | METEOR | CIDEr | SPICE |
|---|---|---|---|---|
| Trained with Cross-Entropy | | | | |
| TOP1-in-2020 [3] | 16.7 | 16.9 | 26.1 | 10.6 |
| BUTD [24] | 18.4 | 17.4 | 29.5 | 11.3 |
| Ours PTM | **19.7** | 18.4 | 33.8 | 12.3 |
| Ours CE-PTM | 19.6 | **18.8** | **34.5** | **12.7** |
| Trained with Reinforcement Learning | | | | |
| TOP1-in-2020 [3] | 17.4 | 16.9 | 28.3 | 10.6 |
| BUTD [24] | 19.4 | 17.9 | 31.7 | 11.4 |
| Ours PTM | 21.3 | 18.8 | 35.4 | **12.7** |
| Ours CE-PTM | **21.4** | **19** | **35.8** | **12.7** |
| Hybrid reranking | | | | |
| BUTD [24] | 20.2 | 18.5 | 34.7 | 12.2 |
| PTM | 21.4 | 19.0 | 37.1 | 13.0 |
| PTM+CE-PTM | **21.6** | **19.3** | 38.1 | 13.3 |
| BUTD+PTM+CE-PTM | 21.5 | **19.3** | **38.5** | **13.4** |

Finally, we submit four runs as follows, and their final evaluation results on TRECVID VTT 2021 dataset are shown in Table 3. Our system ranks 1st place on METEOR, CIDEr, SPICE, and STS metrics and 2nd place on BLEU4.

- Run 4: Our single best model.
- Run 3: Ensemble of the BUTD models.
- Run 2: Ensemble of the PTM and CE-PTM models.
- Run 1: Ensemble of run2 and run3 by captions reranking.

Table 3: Results of the submitted four runs on TRECVID VTT 2021 dataset.

| Runs | BLEU@4 | METEOR | CIDEr | SPICE | STS |
|---|---|---|---|---|---|
| 4 | 3.9 | 31.6 | 33.6 | 11.9 | 44.1 |
| 3 | 3.7 | 31.1 | 32.4 | 11.6 | 44.2 |
| 2 | **4.7** | **32.7** | 35.9 | **12.7** | **45.7** |
| 1 | 4.6 | 32.5 | **36.0** | 12.6 | 45.6 |

## 2 Fill-in-the-Blanks

### 2.1 Approach

As described in [25], the Fill-in-the-Blanks subtask requires the model to complete the video description with a blank based on the video content. Since no training data is provided in this challenge, we automatically build pseudo (video, caption with blank) pairs based on video captioning datasets as mentioned in the Description Generation subtask. Note that a blank represents a single concept but not necessarily a single word. We extract verb and noun phrases with article, adjective, adverb, and preposition in a caption to form blanks. Since the PTM model described in the Description Generation subtask can recover the masked words, we consider using PTM as well for the Fill-in-the-Blanks subtask. Specifically, we propose four approaches to generate the blank phrases based on PTM, including Non-autoregressive Mask Generation (NMG), Auto-regressive Mask Generation (AMG), LSTM Decoder Generation (LDG), and Transformer Decoder Generation (TDG). The illustrations of these approaches are shown in Figure 2.

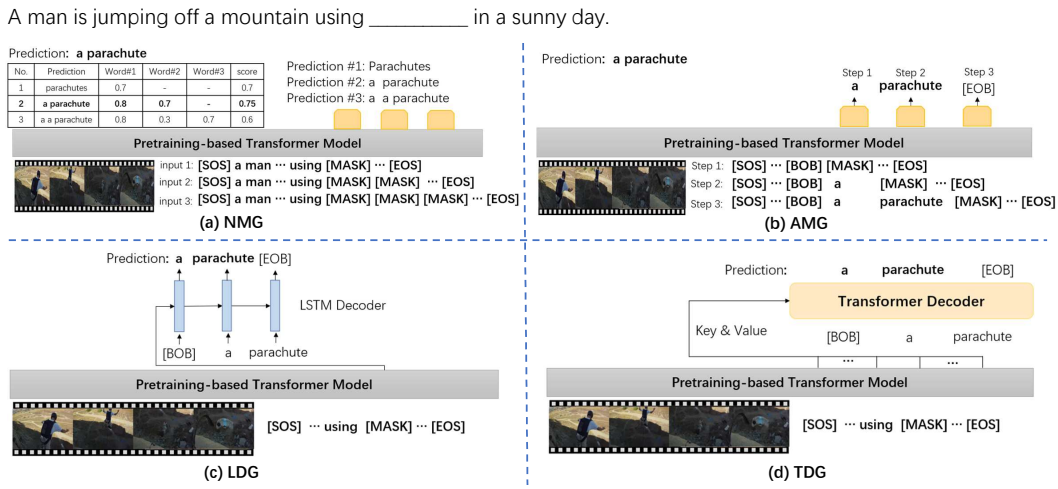


Figure 2: The illustrations of blank generation approaches proposed in this report.

**Non-autoregressive Mask Generation (NMG).** As shown in Figure 2(a), in NMG, the blank in a caption is treated as successive [MASK] tokens and finetune the PTM with the pseudo pairs using VMLM objective. We find that the pseudo blanks are usually shorter than 3 words. Therefore, we feed 1-3 successive [MASK] tokens and generate three phrases with the length of 1, 2, and 3 respectively. We perform mean pooling to aggregate the word-level confident scores into a blank-level one for each prediction. The most confident prediction is treated as the final prediction.

**Auto-regressive Mask Generation (AMG).** In NMG, the multiple words in the blank are simultaneously predicted without intra-context. To exploit the intra-context in the blank, we propose AMG that generates the blank phrase in an auto-regressive manner similar to the description generation strategy described in Section 1. The process of AMG is illustrated in Figure 2(b). Different from generating video captions that only need the previous information to generate the current word, the context information before the blank is also critical for the Fill-in-the-Blanks task. The current word generation in the blank depends on (1) context words before the blank (2) the words in the blank before the current position and (3) context words after the blank, without the words after the current position in the blank. Therefore, if we constrain the attention mask of the current word as in Section 1, the positional embeddings of the context words behind after blank will disclose the length information of the blank implicitly. To address this problem, we do not constrain the attention mask anymore, and just remove the words in the blank after the current position. More precisely, in the fine-tuning stage, we feed a random prefix of the pseudo blank appended with a [MASK] token into the context words as the input of the model. The model is then asked to predict the masked word. During inference, we replace the blank with a special start token [BOB] and a [MASK] token to generate the first blanked word. Once a word is predicted, the [MASK] token in the blank is replaced by the predicted word, and a new [MASK] is appended to generate the next word until the special

end token [EOB] is generated. Note that once we append an [MASK] in the blank, the positional embeddings of the context words behind the blank need to add one during inference.

**LSTM Decoder Generation (LDG).** The NMG and AMG approaches both directly predict the blank solely based on the PTM encoder. Considering that the blanks are phrases with variable lengths, we further explore using LSTM as an additional decoder to generate the blank phrase. As shown in Figure 2(c), we replace the blank with one [MASK] token and feed its hidden state $h_{[\mathrm{MASK}]}$ from the PTM encoder into an LSTM model as the initial hidden state.

We train the LSTM to generate blank phrase with the cross entropy loss as follows:

$$h_0 = h_{[MASK]} \tag{8}$$

$$h_i = \mathrm{LSTM}(y_{i-1}, h_{i-1}; \theta_l) \tag{9}$$

$$p(y_i|y_{<i}) = \mathrm{softmax}(h_i W_{ld}^T) \tag{10}$$

$$L_{xe} = -\frac{1}{N_b}\sum_{i=0}^{N_b} \log p(y_i|y_{<i}) \tag{11}$$

where $y_i$ is the $i$-th word embedding of the blank phrase, $\theta_l$ denotes all the parameters of LSTM, $W_{ld}^T$ is the logit matrix and $N_b$ is the length of the blank phrase.

**Transformer Decoder Generation (TDG).** Besides the LSTM Decoder, we also explore to generate blank phrases with Transformer Decoder [14]. Similar to the LDG, TDG also replace the blank phrase with one [MASK] token as shown in Figure 2(d). The whole textual hidden states $H_t = [h_1^t, ...h_{L_t}^t]$ of the PTM are fed to the Transformer Decoder as the prediction context:

$$Y_i = \mathrm{TransformerDecoderLayer}(Y_{i-1}, H_t, H_t, \theta_i) \tag{12}$$

where $Y_i, i >= 1$ denotes the output of the $i$-th decoder layer and $Y_0 = [y_1, ..., y_{N_b}]$ denotes the word embeddings of the blank. $\theta_i$ denotes the parameters of the $i$-th decoder layer. There are 4 layers in our Transformer Decoder in total. We train the whole model with cross entropy loss as follows:

$$p(y_i|y_{<i}) = \mathrm{softmax}(Y_4 W_{tfd}^T) \tag{13}$$

$$L_{xe} = -\frac{1}{N_b}\sum_{i=0}^{N_b} \log p(y_i|y_{<i}) \tag{14}$$

Considering that the PTM encoder is pre-trained while the Transformer Decoder is optimized from scratch, we further explore to pre-train the Transformer Decoder with PTM for better performance. Specifically, we pre-train the Transformer Decoder with a **Random Blank Filling (RBF)** task. We randomly select 1-5 words in the caption and replace them with a [MASK] token as a blank. The model is asked to predict the blank phrase through the above-mentioned approach. After pre-training, we fine-tune the whole model on our pseudo-blank pairs.

**Hybrid Reranking.** We also adopt the Hybrid Reranking strategy in the Fill-in-the-Blanks subtask. In addition to calculating the visually related score between video and the completed sentence, we also employ a pre-trained language model such as RoBERTa [26] to evaluate the language fluency score of our completed caption. The final reranking score is the weighted sum of visually related score and language fluency score.

### 2.2 Experiments

We conduct experiments to evaluate the above methods on the validation set generated based on TRECVID 2020 video captioning dataset. We use the following two metrics to evaluate the fill-in-the-blanks result:

- **Exactly Match (EM)**: the prediction of the model is considered correct only if it is exactly the same with original words.

Table 4: Pseudo validation on TRECVID VTT 2020.

| Model | EM | F1 |
|---|---|---|
| Single Model | | |
| NMG | 19.4 | 39.5 |
| AMG | 21.5 | 40.8 |
| LDG | 28.5 | 45.0 |
| TDG | 28.9 | 45.1 |
| TDG+RBF | **29.1** | **45.8** |
| Hybrid reranking | | |
| All models | **30.3** | **47.0** |

Table 5: Evaluation results on TRECVID VTT 2021.

| System | Automatic Metrics | | Human Evaluation | |
|---|---|---|---|---|
| | EM | F1 | Average | Average Z |
| human | - | - | **85.4** | **42.0** |
| Run2 | 14.1 | 38.7 | **80.1** | **17.3** |
| Run1 | **15.3** | **40.8** | 79.5 | 13.0 |

- **F1 Score**: calculating the Precision and Recall between the prediction and blank phrase at word-level, then calculating the F1 score as follows:

$$\text{Precision} = \frac{TP}{TP + FP} \tag{15}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{16}$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{17}$$

  where $TP$ represents the number of words in both the prediction and blank phrase, $FP$ represents the number of words only in the prediction and $FN$ represents the number of words only in the blank phrase.

The results are shown in Table 4. The decoder-based generation methods (LDG, TDG) perform significantly better than those without additional decoders (NMG, AMG). Although the pseudo blanks are usually very short, the Transformer Decoder performs slightly better than the LSTM Decoder. Additionally, pretraining the decoder with the RBF task brings improvements.

Our final submission contains two runs as follows, and their final evaluation results on the TRECVID VTT 2021 dataset are shown in Table 5.

- Run 2: The single best model with Transformer Decoder (TDG+RBF).
- Run 1: Ensemble of all generation methods mentioned above via Hybrid Reranking.

As shown in Table 5, there is a gap between Automatic Metrics and Human Evaluation. Similar to the validation results, Run1 performs better on two Automatic Metrics, while Run2 has better Human Evaluation scores. It suggests that Exactly Match and F1 Score are not perfect evaluation metrics due to the diversity of blank phrases. Finally, our system ranked 1st on Human Evaluation.

## 3 Conclusions

In this report, we present our systems for the two subtasks of Video to Text Description (VTT) in the TRECVID 2021 challenge. For the Description Generation subtask, we build a Concept Enhanced Pretraining-based Transformer Model (CE-PTM) and finetune it with the same architecture

to generate video descriptions. Hybrid reranking is employed to ensemble different models according to the visual relevancy of generated descriptions. For the Fill-in-the-Blanks subtask, we propose 4 generation methods based on the PTM pre-trained in the Description Generation subtask. Our system finally achieves the best results on METEOR, CIDEr, SPICE, and STS metrics for Description Generation subtask, and the best performance on Human Evaluation for Fill-in-the-Blanks subtask in the TRECVID 2021 challenge.

## References


[1] George Awad, Asad A. Butt, Keith Curtis, Yooyoung Lee, Jonathan Fiscus, Afzal Godil, Andrew Delgado, Jesse Zhang, Eliot Godard, Lukas Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quénot. Trecvid 2020: comprehensive campaign for evaluating video retrieval tasks across multiple application domains. In *Proceedings of TRECVID 2020*. NIST, USA, 2020.

[2] George Awad, Asad A. Butt, Keith Curtis, Jonathan G. Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas L. Diduch, Jeffrey Liu, Alan F. Smeaton, Yvette Graham, Gareth J. F. Jones, Wessel Kraaij, and Georges Quénot. TRECVID 2020: A comprehensive campaign for evaluating video retrieval tasks across multiple application domains. *CoRR*, abs/2104.13473, 2021.

[3] Yida Zhao, Yuqing Song, Shizhe Chen, and Qin Jin. Ruc_aim3 at trecvid 2020: Ad-hoc video search & video to text description. TRECVID, 2020.

[4] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[5] Huaishao Luo, Lei Ji, Botian Shi, Haoyang Huang, Nan Duan, Tianrui Li, Xilin Chen, and Ming Zhou. Univilm: A unified video and language pre-training model for multimodal understanding and generation. *CoRR*, abs/2002.06353, 2020.

[6] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, Yejin Choi, and Jianfeng Gao. Oscar: Object-semantics aligned pre-training for vision-language tasks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 121–137, Cham, 2020. Springer International Publishing.

[7] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020, 2021.

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[9] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[10] Du Tran, Heng Wang, Matt Feiszli, and Lorenzo Torresani. Video classification with channel-separated convolutional networks. In *IEEE International Conference on Computer Vision*, 2019.

[11] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[12] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[13] Deepti Ghadiyaram, Du Tran, and Dhruv Mahajan. Large-scale weakly-supervised pre-training for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[14] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. 2018.

[16] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, 2014.

[17] Luowei Zhou, Hamid Palangi, Lei Zhang, Houdong Hu, Jason J. Corso, and Jianfeng Gao. Unified vision-language pre-training for image captioning and vqa, 2019.

[18] Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[19] Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jarret Ross, and Vaibhava Goel. Self-critical sequence training for image captioning. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, 2017.

[20] Fartash Faghri, David J. Fleet, Jamie Ryan Kiros, and Sanja Fidler. Vse++: Improving visual-semantic embeddings with hard negatives. In *British Machine Vision Conference*, 2018.

[21] Yuncheng Li, Yale Song, Liangliang Cao, Joel Tetreault, Larry Goldberg, Alejandro Jaimes, and Jiebo Luo. Tgif: A new dataset and benchmark on animated gif description. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[22] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[23] Wang Xin, Wu Jiawei, Chen Junkun, Li Lei, Wang Yuan-Fang, and Wang William Yang. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *IEEE International Conference on Computer Vision*, 2019.

[24] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[25] George Awad, Asad A. Butt, Keith Curtis, Jonathan Fiscus, Afzal Godil, Yooyoung Lee, Andrew Delgado, Jesse Zhang, Eliot Godard, Baptiste Chocot, Lukas Diduch, Jeffrey Liu, Yvette Graham, Gareth J. F. Jones, , and Georges Quénot. Evaluating multiple video understanding and retrieval tasks at trecvid 2021. In *Proceedings of TRECVID 2021*. NIST, USA, 2021.

[26] 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, 07 2019.