HERO: Hierarchical Encoder for Video+Language Omni-representation Pre-training

Linjie Li[*], **Yen-Chun Chen**[*], **Yu Cheng**, **Zhe Gan**, **Licheng Yu**, **Jingjing Liu**
Microsoft Dynamics 365 AI Research
{lindsey.li, yen-chun.chen, yu.cheng, zhe.gan, licheng.yu, jingjl}@microsoft.com

Abstract

We present HERO, a Hierarchical EncodeR for Omni-representation learning, for large-scale video+language pre-training. HERO encodes multimodal inputs in a hierarchical fashion, where *local* textual context of a video frame is captured by a Cross-modal Transformer via multimodal fusion, and *global* video context is captured by a Temporal Transformer. Besides standard Masked Language Modeling (MLM) and Masked Frame Modeling (MFM) objectives, we design two new pre-training tasks: (i) Video-Subtitle Matching (VSM), where the model predicts both global and local temporal alignment; and (ii) Frame Order Modeling (FOM), where the model predicts the right order of shuffled video frames. Different from previous work that mostly focused on cooking or narrated instructional videos, HERO is jointly trained on HowTo100M and large-scale TV show datasets to learn complex social scenes, dynamics backdrop transitions and multi-character interactions. Extensive experiments demonstrate that HERO achieves new state of the art on both text-based video moment retrieval and video question answering tasks across different domains.

1 Introduction

Inspired by BERT (Devlin et al., 2019), large-scale multimodal pre-training has prevailed in the arena of vision-and-language research (Lu et al., 2019a; Tan and Bansal, 2019; Chen et al., 2019b). However, most existing models are tailored for static images, not dynamic videos.VideoBERT (Sun et al., 2019b) was the first to apply BERT to learn joint embedding for video-text pairs. But as only discrete tokens are used to represent video frames, rich video frame features are not fully utilized. To remedy this, CBT (Sun et al., 2019a) uses a contrastive loss but still mainly for video representation learning alone, with text input only considered as side information. UniViLM (Luo et al., 2020) takes a step further and considers both understanding (*e.g.*, text-based video retrieval) and generation (*i.e.*, video captioning) tasks.

Several limitations cast constraints on the scope of existing models. (i) Most model designs are direct adaptation of BERT, without considering the unique characteristics of video+text input. Subtitle sentences and visual frames are usually concatenated, while losing the temporal alignment between different modalities. (ii) Pre-training tasks are directly borrowed from image+text pre-training, without exploiting the sequential nature of video input. (iii) Compared to diverse image domains, video datasets investigated in existing models are restricted to cooking or narrated instructional videos (Miech et al., 2019), excluding video sources that contain dynamic scene transitions and multi-character interactions.

To address these challenges, we present a new video-and-language large-scale pre-training approach - HERO (**H**ierarchical **E**ncode**R** for **O**mni-representation learning). As illustrated in Figure 1, HERO takes as input video clip frames and their accompanying subtitle sentences[1]. Instead of adopting a flat BERT-like encoder, HERO encodes multimodal inputs in a hierarchical fashion, with (i) a *Cross-modal* Transformer to fuse a subtitle sentence and its accompanying local video frames, followed by (ii) a *Temporal* Transformer to obtain a sequentially contextualized embedding for each video frame, using all the surrounding frames as global context. The proposed hierarchical model first absorbs visual and textual local context on frame level, which is then transferred to a global clip-level temporal context. Experiments show that this novel model design achieves better per-

[*] Equal contribution.

[1] ASR can be applied when subtitles are unavailable.

formance than a flat BERT-like architecture.

Four pre-training tasks are designed for HERO: (i) Masked Language Modeling (MLM); (ii) Masked Frame Modeling (MFM); (iii) Video-Subtitle Matching (VSM); and (iv) Frame Order Modeling (FOM). Compared to previous work, the key novelty is VSM and FOM, which encourages explicit temporal alignment between multimodalities as well as full-scale exploitation of the sequential nature of video input. In VSM, the model considers not only global alignment, by predicting whether a subtitle matches the input video clip; but also local temporal alignment, by retrieving the moment where the subtitle should be localized in the video clip. In FOM, we randomly select and shuffle a subset of video frames, and the model is trained to restore their original order. Extensive ablation studies demonstrate that both VSM and FOM play a critical role in video+language pre-training.

To empower the model with richer knowledge such as contextual understanding of dynamic social interactions between multi-characters and dramatic scene/event evolvement, we jointly train HERO on two diverse datasets: HowTo100M dataset (containing 22k narrated instructional videos) (Miech et al., 2019) and a large-scale TV dataset (containing 660k TV episodes spanning different genres) (Lei et al., 2018, 2019, 2020; Liu et al., 2020). Compared to factual and instructional descriptions in HowTo100M, the TV dataset contains more complex plots that require comprehensive interpretation of human emotions, social relations and causal relations of events, which makes it a valuable supplement to HowTo100M and a closer approximation to real-life scenarios.

Previous models pre-trained on HowTo100M are evaluated on YouCook2 (Zhou et al., 2018a) and MSR-VTT (Xu et al., 2016) datasets. YouCook2 focuses on cooking videos only, and the captions in MSR-VTT are very simple. To evaluate our model on more challenging benchmarks, we collect two new datasets on video moment retrieval (*HowTo100M-R*) and question answering (*HowTo100M-QA*). We also evaluate on TVR (Lei et al., 2020) and TVQA (Lei et al., 2018), with extensive ablation studies on pre-training settings.

Our main contributions are summarized as follows. (i) We present HERO, a hierarchical Transformer-based encoder for video+language representation learning. (ii) We propose new pre-training tasks VSM and FOM, which complements MLM and MRM objectives by better capturing temporal alignment between multimodalities in both global and local contexts. (iii) Different from previous work that mainly relies on HowTo100M, we include additional large-scale TV show datasets for pre-training, encouraging the model to learn from richer and more divserse visual content. (iv) We also collect two new datasets based on HowTo100M for video moment retrieval/QA, and will release the new benchmarks to foster future studies. HERO achieves new state of the art across all the evaluated tasks.

2 Related Work

2.1 Model Pre-training

Since the birth of BERT (Devlin et al., 2019), there has been continuing advancement in language model pre-training, such as XLNet (Yang et al., 2019), RoBERTa (Liu et al., 2019), ALBERT (Lan et al., 2019), UniLM (Dong et al., 2019), and T5 (Raffel et al., 2019), which epitomizes the superb power of large-scale pre-training. Satellited around BERT, there are also studies on model compression (Sun et al., 2019c; Jiao et al., 2019; Shen et al., 2019) and extension from understanding to generation (Chen et al., 2019a; Clinchant et al., 2019; Wang and Cho, 2019).

Branching out from language processing towards multimodality, there also emerged subsequent studies in vision+language space. Pioneering works such as ViLBERT (Lu et al., 2019a) and LXMERT (Tan and Bansal, 2019) propose to encode image and text modalities by two separate Transformers, with a third Transformer for later multimodal fusion. Compared to this two-stream architecture, VL-BERT (Su et al., 2019), Unicoder-VL (Li et al., 2019a), B2T2 (Alberti et al., 2019), VisualBERT (Li et al., 2019b), and UNITER (Chen et al., 2019b) advocate single-stream architecture, where image and text signals are fused together in early stage. More recently, ViLBERT is enhanced by multi-task learning (Lu et al., 2019b), Oscar (Li et al., 2020) enhances pre-training with image tags, and Pixel-BERT (Huang et al., 2020) proposes to align image pixels (instead of bottom-up features (Anderson et al., 2018)) with text.

Contrast to the boom in other areas, video+language pre-training is still in its infancy. VideoBERT (Sun et al., 2019b), CBT (Sun et al., 2019a) and UniViLM (Luo et al., 2020) are the only existing works exploring this space. In this paper, we aim to propel video+language

omni-presentation learning in four dimensions: (i) better model architecture design; (ii) better pre-training task design; (iii) diversification of training corpora; and (iv) new high-quality benchmarks for downstream evaluation.

2.2 Video+Language Tasks

Text-based video moment retrieval is one of the most popular video+language tasks currently studied. Anne Hendricks et al. (2017) and Gao et al. (2017) introduce the task of Single Video Moment Retrieval (SVMR), which aims at retrieving a moment from a single video via a natural language query. Escorcia et al. (2019) extends SVMR to Video Corpus Moment Retrieval (VCMR), extending searching pool from single video to large video corpus. TVR (Lei et al., 2020) defines a new task: Video-Subtitle Corpus Moment Retrieval, which provides temporally aligned subtitle sentences along with the videos as inputs. For this new task, XML (Lei et al., 2020) is proposed to compute similarity scores between the query and each modality separately (visual frames, subtitles) and then sum them together for final prediction.

Video question answering (QA) aims to predict answers to natural language questions given a video as context. Some tasks collect QA pairs based on one modality only. For example, MovieFIB (Maharaj et al., 2017) focuses on visual concepts, MovieQA (Tapaswi et al., 2016) is based on text summaries, and TGIF-QA(Jang et al., 2017) uses predefined templates for question generation on short GIFs. TVQA (Lei et al., 2018) designed a more realistic multimodal setting: collecting human-written QA pairs along with their associated video segments by providing both video clips and accompanying subtitles. Later on, Lei et al. (2019) augmented TVQA with frame-level bounding box annotations for spatial-temporal video QA, and introduced the STAGE framework to jointly localize moments, ground objects, and answer questions.

Another popular task is video captioning (Venugopalan et al., 2015; Pan et al., 2016; Gan et al., 2017; Zhou et al., 2018b, 2019), mostly benchmarking on Youtube2Text (Guadarrama et al., 2013), MSR-VTT (Xu et al., 2016), YouCook2 (Zhou et al., 2018a), ActivityNet Captions (Krishna et al., 2017) and VATEX (Wang et al., 2019).

3 Hierarchical Video+Language Encoder

In this section, we introduce the proposed HERO architecture (Sec. 3.1) and explain the four pre-training tasks in detail (Sec. 3.2).

3.1 Model Architecture

Model architecture of HERO is illustrated in Figure 1. HERO takes in the visual frames of a video clip and the textual tokens of subtitle sentences as inputs. First, the inputs are fed into a Video Embedder and a Text Embedder to extract their respective embeddings. Second, HERO computes contextualized video embeddings in a hierarchical fashion. The *local* textual context of each visual frame is captured by a Cross-modal Transformer, while a Temporal Transformer takes *global* video context into consideration. To be more specific, the Cross-modal Transformer computes contextualized multi-modal embeddings between a subtitle sentence and its associated visual frames. The encoded frame embeddings within the whole video clip are then collected, and fed into the Temporal Transformer to obtain the final contextualized video embeddings.

Frame-Subtitle Pairing Given a pair of video clip and its associated subtitle, we first extract a sequence of visual frames $\mathbf{v} = \{v_i\}_{i=1}^{N_v}$ at a fixed frame rate (N_v is the number of visual frames in a video clip). The subtitle is parsed into sentences $\mathbf{s} = \{s_i\}_{i=1}^{N_s}$ (N_s is the number of sentences in each subtitle). Note that $N_v \neq N_s$ in most cases, since a subtitle sentence may last for several visual frames. We then align the subtitle sentences temporally with the visual frames. Specifically, for each subtitle sentence s_i, we pair it with a sequence of visual frames whose timestamps overlap with the subtitle timestamp, and denote these visual frames as $\mathbf{v}_{s_i} = \{v_{s_i}^j\}_{j=1}^{K}$ (K is the number of overlapping frames with s_i). In the case that multiple sentences overlap with the same visual frame, we always pair the frame with the one with maximal temporal Intersection over Union (tIoU) to avoid duplication. It is possible that a subtitle sentence is not paired with any visual frame, and in this case, we concatenate it to the neighboring sentences to avoid information loss.

Input Embedder For *Text Embedder*, we follow Liu et al. (2019) and tokenize a subtitle sentence s_i into a sequence of WordPieces (Wu et al., 2016) sub-word tokens, *i.e.*, $\mathbf{w}_{s_i} = \{w_{s_i}^j\}_{j=1}^{L}$ (L is the number of tokens in s_i). The final representation for each sub-word token is obtained via summing up its token embedding and position embedding, followed by another layer normalization (LN) layer.

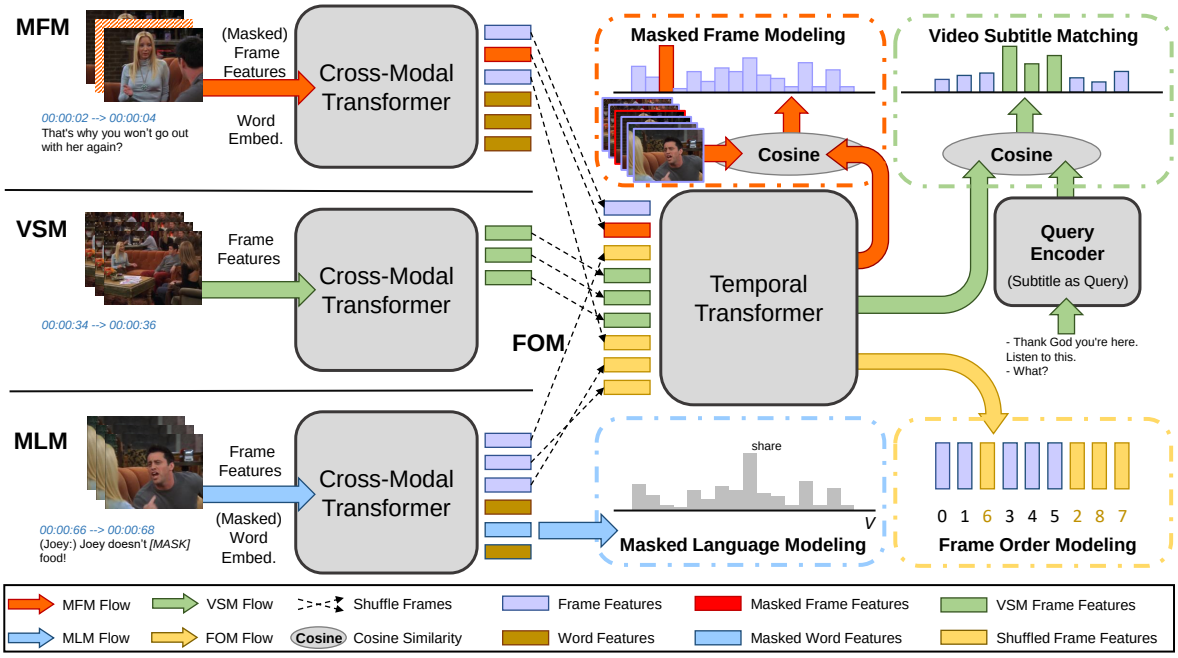

Figure 1: Overview of HERO model (best viewed in color), consisting of Cross-Modal Transformer and Temporal Transformer, learned via four pre-training tasks hierarchically. Initial frame features are obtained by SlowFast and ResNet feature extractors, and initial word embeddings are learned via an embedding layer initialized from RoBERTa. During training, we sample one task per mini-batch to prevent different tasks from corrupting each others' inputs. Sec. 3 provides more detailed descriptions on model architecture and each pre-training task.

For *Video Embedder*, we first use ResNet (He et al., 2016) pre-trained on ImageNet (Deng et al., 2009) and SlowFast (Feichtenhofer et al., 2019) pre-trained on Kinetics (Kay et al., 2017) to extract 2D and 3D visual features for each video frame. The 2D and 3D features are concatenated as our visual features, which are fed through a fully-connected (FC) layer to be projected into the same lower-dimensional space as token embeddings. Since video frames are sequential, their position embeddings can be calculated in the same way as in Text Embbedder. The final embedding of a visual frame is obtained by summing up FC outputs and position embeddings and then passing through a LN layer. In a summary, after Input Embedder, the token embeddings and visual frame embeddings corresponding to $\mathbf{w}_{s_i}$ and $\mathbf{v}_{s_i}$ are denoted as $\mathbf{W}^{emb}_{s_i} \in \mathbb{R}^{L\times d}$ and $\mathbf{V}^{emb}_{s_i} \in \mathbb{R}^{K\times d}$ (d is hidden size), respectively.

Cross-modal Transformer To utilize the inherent alignment between subtitles and video frames, for each subtitle sentence s_i, we first learn contextualized embeddings between the corresponding tokens $\mathbf{w}_{s_i}$ and its associated visual frames $\mathbf{v}_{s_i}$ through cross-modal attention. Inspired by the recent success (Chen et al., 2019b; Lu et al., 2019a) of using Transformer (Vaswani et al., 2017) for multimodal fusion, we also use a multi-layer Transformer here. The outputs from Cross-modal Transformer is a sequence of contextualized embeddings for each subtitle token and each video frame:

$$\mathbf{V}^{cross}_{s_i}, \mathbf{W}^{cross}_{s_i} = f_{cross}(\mathbf{V}^{emb}_{s_i}, \mathbf{W}^{emb}_{s_i}), \quad (1)$$

where $f_{cross}(\cdot,\cdot)$ denotes the Cross-modal Transformer, $\mathbf{V}^{cross}_{s_i} \in \mathbb{R}^{K\times d}$ and $\mathbf{W}^{cross}_{s_i} \in \mathbb{R}^{L\times d}$.

Temporal Transformer After collecting all the visual frame embeddings $\mathbf{V}^{cross} = \{\mathbf{V}^{cross}_{s_i}\}_{i=1}^{N_s} \in \mathbb{R}^{N_v\times d}$ from the output of Cross-modal Transformer, we use another Transformer as temporal attention to learn contextualized video embeddings from the global context of a video clip. To avoid losing positional information, we use residual connection (He et al., 2016) to add back $\mathbf{V}^{emb} \in \mathbb{R}^{N_v\times d}$. The final contextualized video embeddings are calculated as:

$$\mathbf{V}^{temp} = f_{temp}(\mathbf{V}^{emb} + \mathbf{V}^{cross}), \quad (2)$$

where $f_{temp}(\cdot)$ denotes the Temporal Transformer, and $\mathbf{V}^{temp} \in \mathbb{R}^{N_v\times d}$. Compared with a flat BERT-like encoder, which directly concatenate all the textual tokens and visual frames as model inputs,

the proposed model effectively utilizes the temporal alignment between subtitle sentences and video frames for multi-modal fusion in a more fine-grained manner. In the experiments, we show our model design far outperforms a flat BERT-like baseline.

3.2 Pre-training Tasks

We introduce four main tasks to pre-train our model: Masked Language Modeling (MLM), Masked Frame Modeling (MFM) (with two variants), Video-Subtitle Matching (VSM), and Frame Order Modeling (FOM)[2]. As shown in Figure 1, MFM and MLM are in analogy to BERT (Devlin et al., 2019). Word masking is realized by replacing the word with a special token `[MASK]`, and frame masking by replacing the visual feature vector of a frame with zeros. Following Chen et al. (2019b), we only mask one modality each time while keeping the other modality intact. VSM is designed to learn both *local* alignment (between visual frames and a subtitle sentence) and *global* alignment (between a video clip and a sequence of subtitle sentences). FOM is designed to model sequential characteristics of visual clips, by learning the original order of randomly reordered frames.

3.2.1 Masked Language Modeling

The inputs for MLM include: (i) sub-word tokens from the i-th subtitle sentence $\mathbf{w}_{s_i}$; (ii) visual frames $\mathbf{v}_{s_i}$ aligned with $\mathbf{w}_{s_i}$; and (iii) mask indices $\mathbf{m} \in \mathbb{N}^M$.[3]

In MLM, we randomly mask out input words with a probability of 15%, and replace the masked tokens $\mathbf{w}_{s_i}^{\mathbf{m}}$ with special tokens `[MASK]`[4]. The goal is to predict these masked words based on the observation of their surrounding words $\mathbf{w}_{s_i}^{\setminus \mathbf{m}}$ and the visual frames aligned with the sentence $\mathbf{v}_{s_i}$, by minimizing the negative log-likelihood:

$$\mathcal{L}_{\text{MLM}}(\theta) = -\mathbb{E}_D \log P_\theta(\mathbf{w}_{s_i}^{\mathbf{m}} | \mathbf{w}_{s_i}^{\setminus \mathbf{m}}, \mathbf{v}_{s_i}) \,, \quad (3)$$

where θ denotes the trainable parameters, and each pair $(\mathbf{w}_{s_i}, \mathbf{v}_{s_i})$ is sampled from the whole training set D.

[2]For pre-training, we randomly sample one task for each mini-batch, and train on only one objective per SGD update.

[3]$\mathbb{N}$ is a natural number, M is the number of masked tokens, and $\mathbf{m}$ is the set of masked indices.

[4]Following BERT, we decompose the 15% randomly masked-out words into 10% random words, 10% unchanged, and 80% `[MASK]`.

3.2.2 Masked Frame Modeling

Similar to MLM, we also sample frames and mask their visual features with a probability of 15%. However, the difference is that MLM is performed on a local context (*i.e.*, the output of Cross-modal Transformer), while MFM is performed on the global context (*i.e.*, the output of Temporal Transformer). The model is trained to reconstruct masked frames $\mathbf{v}_{\mathbf{m}}$, given the remaining frames $\mathbf{v}_{\setminus \mathbf{m}}$ and all the subtitle sentences $\mathbf{s}$. The visual features of masked frames are replaced by zeros. Unlike textual tokens that are represented as discrete labels, visual features are high-dimensional and continuous, thus cannot be supervised via class likelihood. Instead, we propose two variants for MFM, which share the same objective base:

$$\mathcal{L}_{\text{MFM}}(\theta) = \mathbb{E}_D f_\theta(\mathbf{v}_{\mathbf{m}} | \mathbf{v}_{\setminus \mathbf{m}}, \mathbf{s}) \,. \quad (4)$$

Masked Frame Feature Regression (MFFR) MFFR learns to regress the output on each masked frame $\mathbf{v}_{\mathbf{m}}^{(i)}$ to its visual features. Specifically, we apply an FC layer to convert its output into a vector $h_\theta(\mathbf{v}_{\mathbf{m}}^{(i)})$ of same dimension as the input visual feature $r(\mathbf{v}_{\mathbf{m}}^{(i)})$. Then we apply L2 regression between the two: $f_\theta(\mathbf{v}_{\mathbf{m}} | \mathbf{v}_{\setminus \mathbf{m}}, \mathbf{s}) = \sum_{i=1}^{M} \| h_\theta(\mathbf{v}_{\mathbf{m}}^{(i)}) - r(\mathbf{v}_{\mathbf{m}}^{(i)}) \|_2^2$.

Masked Frame Modeling with Noise Contrastive Estimation (MNCE) Instead of directly regressing the real values of masked visual features, we use the softmax version of the Noise Contrastive Estimation (NCE) loss (Jozefowicz et al., 2016), which has been widely adopted in self-supervised representation learning (Sun et al., 2019a; Hjelm et al., 2019; Oord et al., 2018). NCE loss encourages the model to learn to identify the correct frame (given the context) compared to a set of negative distractors.

Similar to MFFR, we feed the output of the masked frames $\mathbf{v}_{\mathbf{m}}^{(i)}$ into an FC layer to project them into vector $g_\theta(\mathbf{v}_{\mathbf{m}}^{(i)})$. Moreover, we randomly sample frames from the output of unmasked frames as negative distractors $\mathbf{v}_{\mathbf{neg}} = \{\mathbf{v}_{\mathbf{neg}}^{(j)} | \mathbf{v}_{\mathbf{neg}}^{(j)} \in \mathbf{v}_{\setminus \mathbf{m}}\}$, which are also transformed through the same FC layer as $g_\theta(\mathbf{v}_{\mathbf{neg}}^{(j)})$. The final objective minimizes the NCE loss: $f_\theta(\mathbf{v}_{\mathbf{m}} | \mathbf{v}_{\setminus \mathbf{m}}, \mathbf{s}) = \sum_{i=1}^{M} \log \text{NCE}(g_\theta(\mathbf{v}_{\mathbf{m}}^{(i)}) | g_\theta(\mathbf{v}_{\mathbf{neg}}))$.

3.2.3 Video-Subtitle Matching

The inputs to VSM are: (i) a sampled query s_q from all subtitle sentences, (ii) the whole video

clip $\mathbf{v}$, and (iii) the rest subtitle sentences $\mathbf{s}_{\backslash q}$ for the video clip. We expect the model to learn: (i) *local alignment* - the start and end index $y_{st}, y_{ed} \in \{1, ..., N_v\}$, indicating the span of visual frames aligned with the query; and (ii) *global alignment* - which video is the sampled query matched to.

In VSM, we follow XML (Lei et al., 2020) to compute the matching scores between query and visual frames at both local and global levels. Specifically, we extract the output of Temporal Transformer as the final visual frame representation $\mathbf{V}^{temp} \in \mathbb{R}^{N_v \times d}$. The query is fed into Cross-modal Transformer to compute its textual representations $\mathbf{W}_{s_q}^{cross} = f_{cross}(\mathbf{0}, \mathbf{W}_{s_q}^{embed})$. Based on this, we use a query encoder (Lei et al., 2020), consisting of a self-attention layer, two linear layers and a LN layer, to obtain the final query vector $\mathbf{q} \in \mathbb{R}^d$ from $\mathbf{W}_{s_q}^{cross}$.

Local Alignment The local query-video matching score is computed using dot product:

$$S_{local}(s_q, \mathbf{v}) = \mathbf{V}^{temp}\mathbf{q} \in \mathbb{R}^{N_v} . \quad (5)$$

Two trainable 1D convolution filters are applied to the scores, followed by a softmax layer, to generate two probability vectors $\mathbf{p}_{st}, \mathbf{p}_{ed} \in \mathbb{R}^{N_v}$, representing the probabilities of every position being the start and end of the ground-truth span. During training, we sample 15% subtitle sentences as queries for each video, and use the cross-entropy loss to predict the start and end index for local alignment:

$$\mathcal{L}_{local} = -\mathbb{E}_D \log(\mathbf{p}_{st}[y_{st}]) + \log(\mathbf{p}_{ed}[y_{ed}]) ,$$

where $\mathbf{p}[y]$ denotes indexing the y-th element of the vector $\mathbf{p}$.

Note that, XML computes the query-video matching score for each modality separately, and the final matching score is the sum of the two scores. In our HERO model, multi-modal fusion is performed at a much earlier stage.

Global Alignment The global matching score is computed by max-pooling the cosine similarities between each frame and query:

$$S_{global}(s_q, \mathbf{v}) = \max\left(\frac{\mathbf{V}^{temp}}{||\mathbf{V}^{temp}||}\frac{\mathbf{q}}{||\mathbf{q}||}\right) . \quad (6)$$

We use a combined hinge loss $\mathcal{L}_h$ (Yu et al., 2018) over positive and negative query-video pairs. For each positive pair $(s_q, \mathbf{v})$, we replace $\mathbf{v}$ or s_q with one from other samples in the same mini-batch to construct two sets of negative examples: $(s_q, \hat{\mathbf{v}})$ and $(\hat{s}_q, \mathbf{v})$, and the training loss is specified as

$$\begin{aligned}\mathcal{L}_h(S_{pos}, S_{neg}) &= \max(0, \delta + S_{neg} - S_{pos}) ,\\ \mathcal{L}_{global} &= -\mathbb{E}_D[\mathcal{L}_h(S_{global}(s_q, \mathbf{v}), S_{global}(\hat{s}_q, \mathbf{v}))\\ &+ \mathcal{L}_h(S_{global}(s_q, \mathbf{v}), S_{global}(s_q, \hat{\mathbf{v}}))] , \quad (7)\end{aligned}$$

where δ is the margin hyper-parameter. The final loss $\mathcal{L}_{\text{VSM}} = \mathcal{L}_{local} + \lambda\mathcal{L}_{global}$, where λ is a hyper-parameter that balances the above two terms.

3.2.4 Frame Order Modeling

The inputs for FOM are: (i) all subtitle sentences $\mathbf{s}$, (ii) visual frames $\mathbf{v}$, and (iii) the reorder indices $\mathbf{r} = \{r_i\}_{i=1}^R \in \mathbb{N}^R$.[5] We randomly select 15% of the frames to be shuffled, and the goal is to reconstruct their original timestamps, denoted as $\mathbf{t} = \{t_i\}_{i=1}^R$, where $t_i \in \{1, ..., N_v\}$. We formulate FOM as a classification problem, where $\mathbf{t}$ is the ground-truth labels of the reordered frames.

Specifically, reordering happens after the multi-modal fusion of subtitle and visual frames, and is therefore applied to the input of Temporal Transformer. The reordered features are fed into Temporal Transformer to produce reordered visual frame embeddings $\mathbf{V}_{\mathbf{r}}^{temp}$. These embeddings are transformed through an FC layer, followed by a softmax layer to produce a probability matrix $\mathbf{P} \in \mathbb{R}^{N_v \times N_v}$, where each column $\mathbf{p}_i \in \mathbb{R}^{N_v}$ represents the scores of N_v timestamp classes that the i-th timestamp belongs to. The final objective is to minimize the the negative log-likelihood (cross-entropy loss):

$$\mathcal{L}_{\text{FOM}} = -\mathbb{E}_D \sum_{i=1}^{R} \log \mathbf{P}[r_i, t_i] . \quad (8)$$

3.3 Downstream Adaptation

The pre-trained model can be readily adapted to downstream video+language tasks through end-to-end finetuning. Below, we describe the detailed adaptation approach to two popular tasks: (i) text-based video moment retrieval, and (ii) video question answering.

Text-based Video Moment Retrieval The input video clip with its accompanying subtitles is encoded by HERO. The input query is encoded by the query encoder from the VSM pre-training task. We follow the same procedure as in VSM to compute query-video matching scores both locally (frame-level) and globally (clip-level). The model is fine-tuned end-to-end using loss $\mathcal{L}_{\text{VSM}}$.

[5] R is the number of reordered frames, and $\mathbf{r}$ is the set of reorder indices.

Video Question Answering For Video QA, we consider the multiple-choice setting. For each answer candidate, the corresponding QA pair is appended to each of the subtitle sentences and fed into the Cross-modal Transformer to perform early fusion with local textual context. In addition, these QA pairs are also appended to the input of Temporal Transformer to be fused with global video context. We use a simple attention layer to compute the weighted-sum-across-time of the QA-aware frame representations from the Temporal Transformer output. These final QA-aware global representations are then fed through an MLP and softmax layer to obtain the probability score $\mathbf{p}_{ans}^{(i)}$ of all the answers for question i. The training objective is

$$\mathcal{L}_{ans} = -\frac{1}{N}\sum_{i=1}^{N} \log \mathbf{p}_{ans}^{(i)}[y_i]\,, \tag{9}$$

where y_i is the index of the ground-truth answer for question i. When supervision is available[6], we also include the span prediction loss:

$$\mathcal{L}_{span} = -\frac{1}{2N}\sum_{i=1}^{N} (\log \mathbf{p}_{st}^{(i)}[y_i^{st}] + \log \mathbf{p}_{ed}^{(i)}[y_i^{ed}])\,, \tag{10}$$

where $\mathbf{p}_{st}^{(i)}$ and $\mathbf{p}_{ed}^{(i)}$ are the prediction scores of the start and end position, obtained by applying weighted-sum-across-answers attention to the Temporal Transformer output followed by two MLPs and a softmax layer. y_i^{st}, y_i^{ed} are the indices of the ground-truth start and end positions for question i.

4 Experiments

In this section, we describe experiments on different downstream tasks that validate the effectiveness of the representations learned by HERO. Detailed ablation studies also provide in-depth analysis of different pre-training settings.

4.1 Pre-training Datasets

Our pre-training dataset is composed of videos from TV and Howto100M datasets. We exclude all the videos that appeared in the downstream tasks to avoid contamination in evaluation. The full pre-training dataset contains 680k video clips with their accompanying subtitles.

TV Dataset (Lei et al., 2018) was built on 6 popular TV shows across 3 genres: medical dramas, sitcoms and crime shows. It contains 21,793 video clips from 925 episodes. Each video clip is 60-90 seconds long, covering long-range scenes with complex character interactions and social/professional activities. Dialogue for each video clip (in the format of "character name: subtitle") is also provided.

Howto100M Dataset (Miech et al., 2019) was collected from YouTube with mostly instructional videos that teach diverse tasks. It contains 1.22 million videos, with activities falling into 12 categories (e.g., Food & Entertaining, Home & Garden, Hobbies & Crafts). Each video is associated with a narration as subtitles that are either written manually or from an Automatic Speech Recognition (ASR) system. The average duration of videos in Howto100M is 6.5 minutes. We cut the videos into 60-second clips to make them consistent with the TV dataset, and exclude videos in non-English languages. These pre-processing steps result in a subset of 660k video clips, accompanied with English subtitles.

4.2 Data Collection

Existing datasets for video moment retrieval and video QA are built on videos from either a single domain or a single modality. In order to evaluate on datasets not only containing diverse video content but also reflecting multimodalities of videos, we collect two new datasets based on Howto100M as additional benchmarks.

We use Amazon Mechanical Turk (AMT) to collect annotations on Howto100M videos. Figure 2 in Appendix shows the interface for video moment retrieval data collection. We randomly sample 29,843 60-second clips from 9,421 videos and present each clip to the annotators, who are asked to select a video segment containing a single, self-contained scene. After video segments are selected, another group of workers are asked to write captions that describe the displayed segment. Narrations are not provided to workers for some video clips to ensure we include queries that are related to video only. On average, selected video segments are 10-20 seconds long. The length (number of words) of queries is diverse, ranging from 8 to 20.

We also present the selected video segments to another group of AMT workers for QA annotations (interface shown in Figure 3 in Appendix). Each worker is assigned with one video segment and asked to write one question, one correct answer

[6] Some existing Video QA tasks require localizing 'frames of interest' for the question, *e.g.*, TVQA+ (Lei et al., 2019).

Pre-training Data		Pre-training Tasks	TVR			TVQA	Howto100M-R			Howto100M-QA
			R@1	R@10	R@100	Acc.	R@1	R@10	R@100	Acc.
TV	1	MLM	2.92	10.66	17.52	71.25	2.06	9.08	14.45	76.42
	2	MLM + MNCE	3.13	10.92	17.52	71.99	2.15	9.27	14.98	76.95
	3	MLM + MNCE + FOM	3.09	10.27	17.43	72.54	2.36	9.85	15.97	77.12
	4	MLM + MNCE + FOM + VSM	**4.44**	**14.69**	**22.82**	72.75	2.78	10.41	**18.77**	77.54
	5	MLM + MNCE + FOM + VSM + MFFR	**4.44**	14.29	22.37	72.75	2.73	10.12	18.05	77.54
TV & Howto100M	6	MLM + MNCE + FOM + VSM	4.34	13.97	21.78	**74.24**	**2.98**	**11.16**	17.55	**77.75**

Table 1: Evaluation on pre-training tasks and datasets using TVR, TVQA, Howto100M-R and Howto100M-QA validation set as benchmarks. Dark and light grey colors highlight the top and second best results across all the tasks trained with TV Dataset. The best results are in bold. For simplicity, we only report video moment retrieval[7] results for TVR and Howto100M-R.

and 3 wrong answers. Similarly, some narrations are hidden to ensure we include QA pairs that are based on video only and not biased by subtitles. In practice, we observe that human-written negative answers suffer from serious bias (i.e., models can learn to predict the correct answer without even absorbing information from the video or subtitles). Therefore, we use adversarial matching (Zellers et al., 2019) to construct negative answers, by selecting a correct answer (from another question) that is most relevant to the current question. We replace one out of three written negative answers in this way. Detailed statistics about the collected datasets are provided in Appendix.

4.3 Downstream Tasks

To validate the effectiveness of HERO, we evaluate on four different downstream tasks. This subsection describes each task and the corresponding evaluation metrics.

TVR (Lei et al., 2020) is built upon the TV dataset, split into 80% train, 10% val, 5% test-public and 5% test-private. On average, 5 queries were collected for each video clip. Among them, 74.2% of queries are related to video only, 9.1% to text only, and 16.6% to both video and text.

TVQA (Lei et al., 2018) was first introduced along with the TV dataset, where given a video clip and the accompanying subtitles, the goal is to answer a multiple-choice question about the video. Each video clip is annotated with 7 questions and 5 answers per question. The start and end points of relevant moments are also provided for each question. The train, val and test video splits are the same as TVR dataset.

Howto100M-R In total, we have collected 67,542 queries for 29,843 60-second clips from 9,421 videos in HowTo100M, on average 2-3 queries per clip. We split the video clips and its associated queries into 80% train, 10% val and 10% test.

Howto100M-QA is collected under multi-choice QA setting. For the same video clips used in Howto100M-R, each is annotated with 2 questions on average and 4 answers per question. Similar to TVQA, we also provide the start and end points for the relevant moment for each question. We split data into 80% train, 10% val and 10% test.

Evaluation Metrics Text-based Video Moment Retrieval can be decomposed into two sub-tasks: (i) Video Retrieval: retrieve the most relevant video clip described by the query; (ii) Moment Retrieval: given the query, localize the correct moment from the most relevant video clip. Model performance on video moment retrieval is measured on these two sub-tasks. A model prediction is correct if: (i) its predicted video matches the ground-truth (in video retrieval); and (ii) its predicted span has high overlap with the ground-truth (in moment retrieval). Average recall at K (R@K) over all queries is used as the evaluation metric for both TVR and Howto100M-R. Temporal Intersection over Union (tIoU) is also used to measure the overlap between the predicted span and the ground-truth span.[7]

TVQA and Howto100M-QA include 3 sub-tasks: QA on the grounded clip, question-driven moment localization, and QA on the full video clip. We only consider QA on the full video clip, as it is the most challenging setting among the three. Accuracy is used to measure model performance.

4.4 Ablation Study

We analyze the effectiveness of our model design, especially with different combinations of pre-training tasks, through ablation studies over downstream tasks.

Pre-training Tasks and Datasets Table 1 summarizes results on all four downstream tasks under

[7] During evaluation, we apply non-maximal suppression (nms) with threshold 0.5 to the predictions. The average recalls are calculated with tIoU>0.7 after applying nms.

different pre-training settings. To evaluate the effectiveness of each pre-training task, we conduct ablation experiments through pre-training on TV dataset only. Comparing to using MLM only (L1 in Table 1), adding MNCE (L2) shows improvement on all downstream tasks. When MLM, MNCE and FOM are jointly trained (L3), there is a large performance gain in accuracy on TVQA and significant improvement on the two Howto100M downstream tasks. Comparable results are achieved on TVR. This indicates that FOM, which models sequential characteristics of video frames, can effectively benefit downstream tasks that rely on temporal reasoning (such as QA tasks).

The best performance is achieved by MLM + MNCE + FOM + VSM (L4). We observe significant performance lift by adding VSM. The local and global alignments between subtitle and visual frames learned through VSM are especially effective on TVR and Howto100M-R. Adding additional MFFR (L5) achieves slightly worse results. Our observation is that MFFR is competing with (instead of complimentary to) MNCE during pre-training, which renders the effect of adding MFFR negligible.

Lastly, we study the effects of pre-training datasets, by augmenting TV dataset with Howto100M dataset and pre-training HERO with the optimal combination of MLM + MNCE + FOM + VSM. The learned model continues to improve over all tasks except TVR. We hypothesize that the comparable result on TVR is due to the domain difference between the augmented videos and TV dataset.

Model Design To validate the effectiveness of the Cross-modal Transformer in HERO, we compare our model with a Hierarchical Transformer (H-TRM) baseline under two settings: (i) without pre-training[8]; (ii) with optimal pre-training (MLM + MNCE + FOM + VSM) over TV dataset. H-TRM is constructed by simply replacing the Cross-modal Transformer with a RoBERTa model and encoding subtitles only. This way, the inputs to the Temporal Transformer in H-TRM are the summation of initial frame embedding and max-pooled subtitle embeddings from RoBERTa. We also compare HERO with a flat BERT-like encoder (F-TRM) where no pre-training is applied. F-TRM takes as input a single sequence by concatenating the embeddings of visual frames and all subtitle sentences, and encodes them through one multi-layer Transformer, as used in previous pre-training methods.

[8]Model parameters are initialized with RoBERTa weights following Lei et al. (2020).

Pre-training	Model	TVR			TVQA
		R@1	R@10	R@100	Acc.
No[8]	F-TRM	1.99	7.76	13.26	31.80
	H-TRM	2.97	10.65	18.68	70.09
	HERO	2.98	10.65	18.25	70.65
Yes	H-TRM	3.12	11.08	18.42	70.03
	HERO	**4.44**	**14.69**	**22.82**	**72.75**

Table 2: Comparison between a flat BERT-like encoder (F-TRM), a Hierarchical Transfomrer (H-TRM) baseline and HERO using TVR and TVQA validation set as benchmarks. Results in the last two rows are obtained from pre-training the models with MLM + MNCE + FOM + VSM on TV Dataset. For simplicity, we report only video moment retrieval[7] for TVR.

Results are summarized in Table 2. (i) When no pre-training is applied, F-TRM is much worse than HERO on both tasks. H-TRM achieves comparable results to HERO on TVR, but worse on TVQA. Unlike F-TRM, H-TRM and HERO explicitly utilize the inherent temporal alignment between two modalities of videos, which is uniquely important for video+language tasks. (ii) With pre-training, HERO shows significant improvement over H-TRM. Our hypothesis is that with the hierarchical design, HERO can capture cross-modal interactions between visual frames and its local textual context better than H-TRM. Such cross-modality joint understanding of visual and textual contexts is critical for video-based retrieval and QA tasks. (iii) Pre-training lifts HERO performance by a large margin, but not very helpful for H-TRM. These results provide strong evidence that cross-modal interactions and temporal alignments between visual frames and its local textual context learned by HERO are essential for these video+language tasks.

4.5 Comparison with SOTA Models

We compare our model with task-specific state-of-the-art models in Table 3. First, we compare with XML (Lei et al., 2020) on text-based video moment retrieval tasks (TVR and Howto100M-R). Results show that our model consistently outperforms XML on both TVR and Howto100M-R, with or without pre-training.

Second, we compare with SOTA models on video QA tasks (TVQA and Howto100M-QA). Note that for TVQA, STAGE (Lei et al., 2019) is trained with additional supervision on spatial grounding, which requires region-level features

Method	TVR			Howto100M-R			TVQA	Howto100M-QA
	R@1	R@10	R@100	R@1	R@10	R@100	Acc.	Acc.
XML (Lei et al., 2020)	2.70	8.93	15.34	2.06	8.96	13.27	-	-
STAGE (Lei et al., 2019)	-	-	-	-	-	-	70.50	-
HERO w/o pre-training[8]	2.98	10.65	18.42	2.17	9.38	15.65	70.65	76.89
HERO w/ pre-training	**4.34**	**13.97**	**21.78**	**2.98**	**11.16**	**17.55**	**74.24**	**77.75**

Table 3: Results on four downstream tasks: TVR, Howto100M-R, TVQA and Howto100M-QA, compared with task-specific state-of-the-art method: XML for TVR and STAGE for TVQA. Only video moment retrieval[7] results are reported for TVR and Howto100M-R.

for each frame of the video. Results show that without additional supervision on spatial grounding or fine-grained region-level features, our HERO model is able to achieve better performance than STAGE on TVQA dataset. We also observe that pre-training significantly boosts the performance of HERO across TVR, Howto100M-R and TVQA tasks.

On Howto100M-QA, since STAGE was specifically designed to leverage region-level features, we cannot directly apply STAGE. Thus, we only compare model performance of HERO without and with pre-training. Results exhibit consistent pattern observed on other downstream tasks: pre-training achieves better performance than without pre-training. Overall, HERO achieves state-of-the-art results on all four downstream tasks.

5 Conclusion

In this paper, we present a hierarchical encoder for video+language omni-representation pre-training. Our HERO model presents a hierarchical architecture, consisting of Cross-modal Transformer and Temporal Transformer for multi-modal fusion. Novel pre-training tasks are proposed to capture temporal alignment both locally and globally. Pre-trained on two large-scale video datasets, HERO exceeds state of the art by a significant margin when transferred to multiple video-and-language tasks. Two new datasets on text-based video moment retrieval and video QA are introduced to serve as additional benchmarks for downstream evaluation. We consider extension of our model to other video-and-language tasks as future work, as well as developing more well-designed pre-training tasks.

References

Chris Alberti, Jeffrey Ling, Michael Collins, and David Reitter. 2019. Fusion of detected objects in text for visual question answering. *arXiv preprint arXiv:1908.05054*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *CVPR*.

Yen-Chun Chen, Zhe Gan, Yu Cheng, Jingzhou Liu, and Jingjing Liu. 2019a. Distilling the knowledge of bert for text generation. *arXiv preprint arXiv:1911.03829*.

Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2019b. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*.

Stéphane Clinchant, Kweon Woo Jung, and Vassilina Nikoulina. 2019. On the use of bert for neural machine translation. *arXiv preprint arXiv:1909.12744*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*.

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

Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified language model pre-training for natural language understanding and generation. In *NeurIPS*.

Victor Escorcia, Mattia Soldan, Josef Sivic, Bernard Ghanem, and Bryan Russell. 2019. Temporal localization of moments in video collections with natural language. *arXiv preprint arXiv:1907.12763*.

Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *ICCV*.

Zhe Gan, Chuang Gan, Xiaodong He, Yunchen Pu, Kenneth Tran, Jianfeng Gao, Lawrence Carin, and Li Deng. 2017. Semantic compositional networks for visual captioning. In *CVPR*.

Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *CVPR*.

Sergio Guadarrama, Niveda Krishnamoorthy, Girish Malkarnenkar, Subhashini Venugopalan, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2013. Youtube2text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *ICCV*.

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

R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. 2019. Learning deep representations by mutual information estimation and maximization. *ICLR*.

Zhicheng Huang, Zhaoyang Zeng, Bei Liu, Dongmei Fu, and Jianlong Fu. 2020. Pixel-bert: Aligning image pixels with text by deep multi-modal transformers. *arXiv preprint arXiv:2004.00849*.

Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. 2017. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2019. Tinybert: Distilling bert for natural language understanding. *arXiv preprint arXiv:1909.10351*.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. 2017. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *ICCV*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Jie Lei, Licheng Yu, Mohit Bansal, and Tamara Berg. 2018. Tvqa: Localized, compositional video question answering. In *EMNLP*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2019. Tvqa+: Spatio-temporal grounding for video question answering. *arXiv preprint arXiv:1904.11574*.

Jie Lei, Licheng Yu, Tamara L Berg, and Mohit Bansal. 2020. Tvr: A large-scale dataset for video-subtitle moment retrieval. *arXiv preprint arXiv:2001.09099*.

Gen Li, Nan Duan, Yuejian Fang, Daxin Jiang, and Ming Zhou. 2019a. Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. *arXiv preprint arXiv:1908.06066*.

Liunian Harold Li, Mark Yatskar, Da Yin, Cho-Jui Hsieh, and Kai-Wei Chang. 2019b. Visualbert: A simple and performant baseline for vision and language. *arXiv preprint arXiv:1908.03557*.

Xiujun Li, Xi Yin, Chunyuan Li, Xiaowei Hu, Pengchuan Zhang, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. 2020. Oscar: Object-semantics aligned pre-training for vision-language tasks. *arXiv preprint arXiv:2004.06165*.

Jingzhou Liu, Wenhu Chen, Yu Cheng, Zhe Gan, Licheng Yu, Yiming Yang, and Jingjing Liu. 2020. Violin: A large-scale dataset for video-and-language inference. *arXiv preprint arXiv:2003.11618*.

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

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. 2019a. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*.

Jiasen Lu, Vedanuj Goswami, Marcus Rohrbach, Devi Parikh, and Stefan Lee. 2019b. 12-in-1: Multi-task vision and language representation learning. *arXiv preprint arXiv:1912.02315*.

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

Tegan Maharaj, Nicolas Ballas, Anna Rohrbach, Aaron Courville, and Christopher Pal. 2017. A dataset and exploration of models for understanding video data through fill-in-the-blank question-answering. In *CVPR*.

Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. 2019. Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In *ICCV*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *NeurIPS*.

Yingwei Pan, Tao Mei, Ting Yao, Houqiang Li, and Yong Rui. 2016. Jointly modeling embedding and translation to bridge video and language. In *CVPR*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2019. Q-bert: Hessian based ultra low precision quantization of bert. *arXiv preprint arXiv:1909.05840*.

Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. 2019. Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. 2019a. Contrastive bidirectional transformer for temporal representation learning. *arXiv preprint arXiv:1906.05743*.

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

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. 2019c. Patient knowledge distillation for bert model compression. *arXiv preprint arXiv:1908.09355*.

Hao Tan and Mohit Bansal. 2019. Lxmert: Learning cross-modality encoder representations from transformers. In *EMNLP*.

Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. 2016. Movieqa: Understanding stories in movies through question-answering. In *CVPR*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NeurIPS*.

Subhashini Venugopalan, Marcus Rohrbach, Jeffrey Donahue, Raymond Mooney, Trevor Darrell, and Kate Saenko. 2015. Sequence to sequence-video to text. In *ICCV*.

Alex Wang and Kyunghyun Cho. 2019. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*.

Xin Wang, Jiawei Wu, Junkun Chen, Lei Li, Yuan-Fang Wang, and William Yang Wang. 2019. Vatex: A large-scale, high-quality multilingual dataset for video-and-language research. In *ICCV*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *CVPR*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *NeurIPS*.

Licheng Yu, Zhe Lin, Xiaohui Shen, Jimei Yang, Xin Lu, Mohit Bansal, and Tamara L Berg. 2018. Mattnet: Modular attention network for referring expression comprehension. In *CVPR*.

Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *CVPR*.

Luowei Zhou, Yannis Kalantidis, Xinlei Chen, Jason J Corso, and Marcus Rohrbach. 2019. Grounded video description. In *CVPR*.

Luowei Zhou, Chenliang Xu, and Jason J Corso. 2018a. Towards automatic learning of procedures from web instructional videos. In *AAAI*.

Luowei Zhou, Yingbo Zhou, Jason J Corso, Richard Socher, and Caiming Xiong. 2018b. End-to-end dense video captioning with masked transformer. In *CVPR*.

A Appendix

A.1 Data Analysis on Howto100M-R and Howto100M-QA

Data Collection Interface Figure 2 and 3 present the interface we used for collecting Howto100M-R and Howto100M-QA, respectively. For Howto100M-R, the annotator is asked to first select a video segment from the presented video clip using the sliding bar, and then enter a description about the selected video segment in the text box shown at the bottom of Figure 2. For Howto100M-QA, we reuse the selected video segment collected for Howto100M-R. The annotators are asked to write a question, a correct answer and 3 wrong answers in the four text boxes shown in Figure 3.

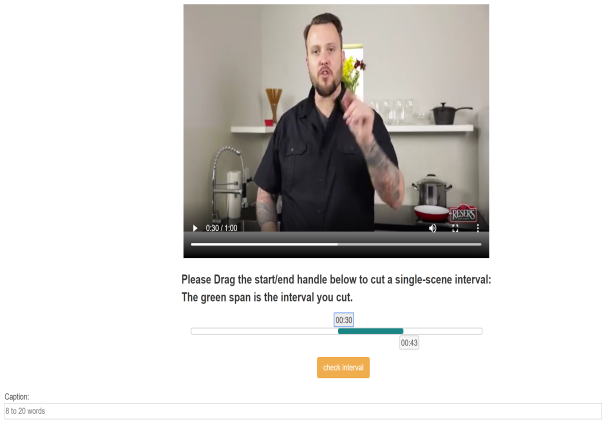

Figure 2: User interface for query annotations. Each worker is provided with a video clip and required to select a single-scene clip from the video, then write a query in the text box.

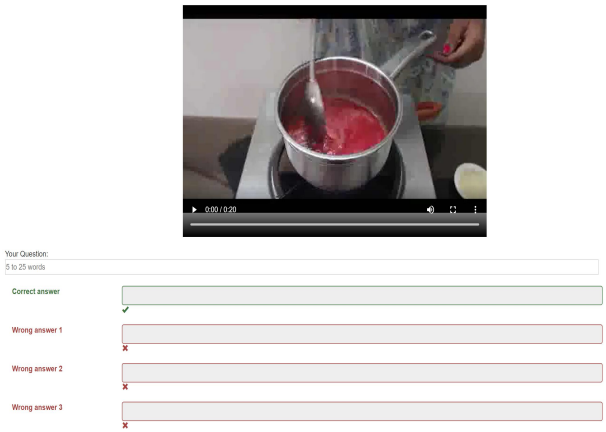

Figure 3: User interface for question/answer annotations. Each worker is provided with a segmented clip and required to write a question with four answers in the text boxes.

Video Segment Length Distribution The length distribution of selected video segment is presented in Figure 4. The lengths of video segments vary from 5 to more than 30 seconds. The majority of them have length less than 15 seconds.

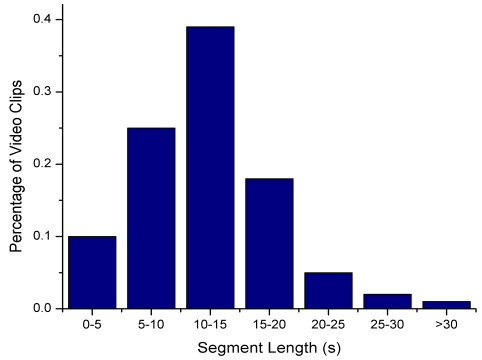

Figure 4: Distribution of video segment lengths.

Howto100M-R Query Length Distribution Figure 5 shows the length (in number of words) distribution of collected queries in Howto100M-R. They have diverse lengths, ranging from 8 to 20.

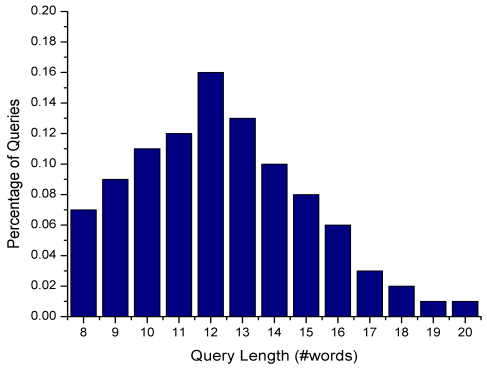

Figure 5: Howto100M-R query length distribution.

Howto100M-QA Question and Answer Distribution Figure 6 and Figure 7 show the length (in number of words) distribution of collected questions and answers in Howto100M-QA. Questions are relatively longer, with more than 10 words on average. Answers are relatively shorter, most of them have less than 7 words.

In addition, we analyze the types of collected question by showing the distribution of their leading words in Figure 8. In total, we collected questions with 7 different types. Majority of them starts with "what", "why" and "when".

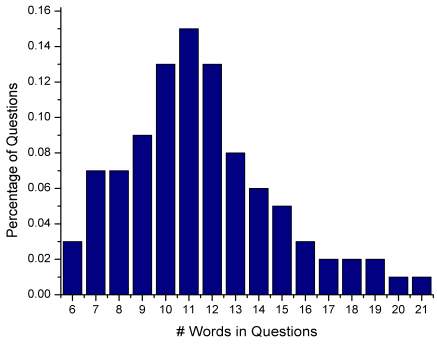

Figure 6: Howto100M-QA question length distribution.

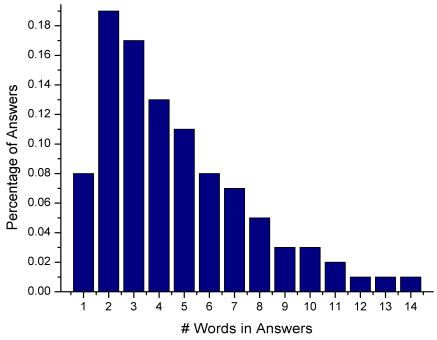

Figure 7: Howto100M-QA answer length distribution.

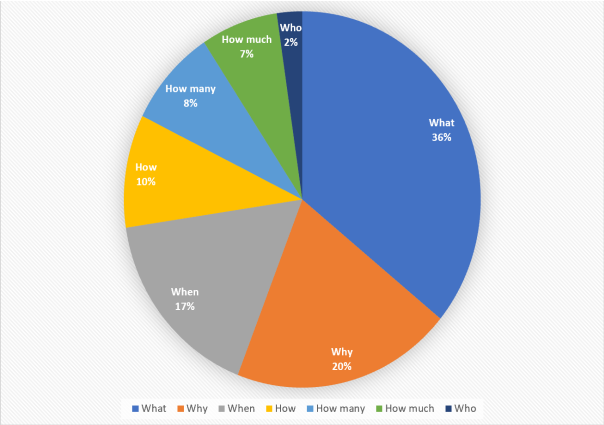

Figure 8: Distribution of questions by their leading words in Howto100M-QA.