# Look Before you Speak: Visually Contextualized Utterances

Paul Hongsuck Seo   Arsha Nagrani   Cordelia Schmid
Google Research
{phseo,anagrani,cordelias}@google.com

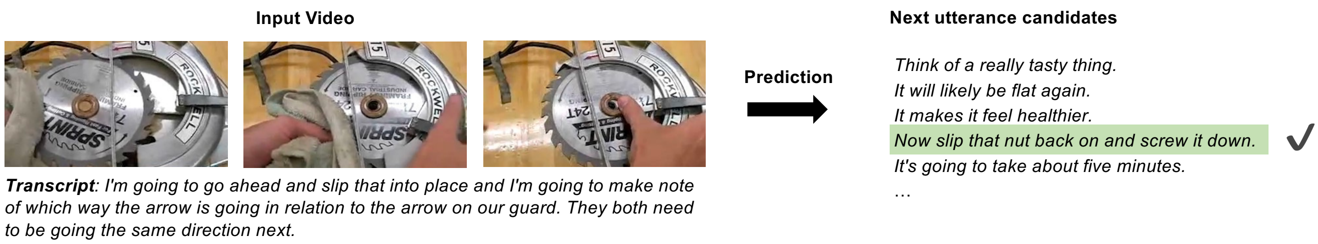



Figure 1: **Visually Contextualised Future Utterance Prediction.** Given an instructional video with paired text and video data, we predict the next utterance in the video using a Co-attentional Multimodal Video Transformer. Our model trained on this task also achieves state-of-the-art performance on downstream VideoQA benchmarks.

## Abstract

*While most conversational AI systems focus on textual dialogue only, conditioning utterances on visual context (when it's available) can lead to more realistic conversations. Unfortunately, a major challenge for incorporating visual context into conversational dialogue is the lack of large-scale labeled datasets. We provide a solution in the form of a new visually conditioned Future Utterance Prediction task. Our task involves predicting the next utterance in a video, using both visual frames and transcribed speech as context. By exploiting the large number of instructional videos online, we train a model to solve this task at scale, without the need for manual annotations. Leveraging recent advances in multimodal learning, our model consists of a novel co-attentional multimodal video transformer, and when trained on both textual and visual context, outperforms baselines that use textual inputs alone. Further, we demonstrate that our model trained for this task on unlabelled videos achieves state-of-the-art performance on a number of downstream VideoQA benchmarks such as MSRVTT-QA, MSVD-QA, ActivityNet-QA and How2QA.*

## 1. Introduction

Imagine that you are cooking an elaborate meal, but forget the next step in the recipe – or fixing your car and uncertain about which tool to pick up next. Developing an intelligent dialogue system[1] that not only emulates human conversation, but also predicts and suggests future actions – not to mention is able to answer questions on complex tasks and topics – has long been a moonshot goal for the AI community. Conversational AI allows humans to interact with systems in free-form natural language, in the same way that we would communicate with one another. This has led to an outpouring of research in NLP focused on conversational agents, ranging from goal-oriented systems for helping with reservations [6, 82] to chit-chat models [46, 69, 87], both of which are found in modern virtual assistants such as Alexa, Google Assistant and Siri.

Such works, however, are limited to linguistic interactions only. In contrast, human interaction in the physical world is facilitated through multiple modalities (*e.g*. verbal, visual, haptic), each modality often complementing the other seamlessly. While doing a task, it is often easier to *show* another person your progress, than to describe it verbally. Hence we argue that a truly intelligent dialogue system would have knowledge of *both* visual and textual contexts before making its next utterance. Unfortunately, a major challenge for incorporating visual context is a lack of suitable data. Most traditional conversational datasets [6, 10, 31, 71] are solely text based, and notoriously difficult to collect, relying on narrowly constructed ontologies [31, 55] and highly specific domains [6, 10]. More importantly, they do not contain visual information of the surrounding physical environment.

In an attempt to incorporate visual context to dialogue systems, the task of visual dialog [15] was proposed, which requires an AI agent to hold a meaningful dialog with humans given an image [15] or a video [38]. In these works, a dialog history and question are artificially created for each image/video in a dataset, and the goal is then to infer context from history and answer the question accurately. Such datasets, while valuable, are created at great manual effort and contain artificially contrived scenarios, where the dialog history is not naturally present in video. Such datasets are also limited in size.

Unlike such works [15, 38], we propose to use online videos

[1] often used interchangeably with the term 'conversational AI'

to learn from naturally co-occurring vision and dialogue in a scalable manner. We note that certain video domains such as *narrated instructional videos* [54, 79] and *lifestyle vlogs* [21, 34] are available in huge numbers (e.g. online on video sharing platforms) and are likely to contain narration explicitly linked to the visual content. Given the availability of high quality ASR, this gives us a large amount of readily available paired visual and textual data. We begin by proposing a future prediction task, where the goal is to predict the next utterance in an instructional video, given both visual and textual contexts (Figure 1). The labels for such a task are freely available from the video itself.

As we show in this work, solving such a task requires knowledge of both visual and textual contexts. Leveraging recent advances in multimodal learning, we do so with a two-stream co-attentional transformer based model, each stream encoding a different modality. Our co-attentional model effectively attends to features within each modality, as well as across modalities through lateral self-attention blocks. We demonstrate that using both visual and textual information leads to a large performance gain over using text alone, and additionally, our two-stream co-attentional model outperforms single stream multimodal models. In addition, we show that our model, trained for this future prediction task, can be transferred to other conversational tasks, achieving state-of-the-art performance on various VideoQA benchmarks.

Concretely, we make the following four contributions: (i) We formulate a future utterance prediction (FUP) task which uses *both* dialogue and vision; (ii) We re-purpose freely available online instructional video datasets to create training and testing benchmarks for this task (HowToFUP); (iii) We propose a new two-stream multimodal video transformer based architecture (CoMVT) which effectively attends jointly over words in text and visual objects and scenes to learn visual-dialogue context; and finally (iv) We show that our model trained on unlabelled instructional videos is also, perhaps surprisingly, able to achieve state-of-the-art performance on a number of downstream vision-language QA datasets, including MSRVTT-QA [85], MSVD-QA [85], ActivityNet-QA [88], and How2QA [47].

## 2. Related Work

**Future Utterance Prediction.** Predicting future utterances from textual data alone has been widely explored in the NLP community for conversational AI systems. Approaches include hand-crafted rules [17, 80], example-based agents [39, 44] and modern neural networks [70, 72], and aim to generate realistic responses for goal-oriented dialog systems or chatbots. Future prediction has also been used as an unsupervised pretraining objective for text corpora, *e.g.* next sentence prediction in BERT [19]. Unlike these works, we focus on jointly learning from visual context as well as text. Related to our work is the task of scene-aware dialog prediction [2, 33], where the goal is to answer questions grounded to a video clip input, given a manually created dialog history. A number of works show promising results on this task [14, 32, 42, 45, 48], however they rely on manually created VQA datasets.

**Vision and Language Tasks.** Popular vision and language tasks include visual question answering [4, 5, 22, 28, 58, 61], visual dialog [15, 16, 36, 40, 67], visual captioning [56, 59, 68, 73, 86], visual grounding [18, 22, 57, 83] and video-text retrieval [7, 25, 49, 53, 63]. There have also been attempts to use transcribed speech in videos as a source of weak supervision [52, 60, 76, 77], where the goal is to learn a good visual encoder, and consequently such works are largely evaluated only on downstream tasks that involve unimodal video frame inputs. In contrast, we learn an encoder that can effectively learn to *co-attend* to both vision and text, and is useful for downstream tasks that involve both modalities.

**Multimodal Vision-Text Architectures.** A large number of multimodal architectures focus on late fusion of modalities, with popular choices being summation, concatenation and canonical correlation analysis [9, 74, 78]. [47] encodes multimodal inputs in a hierarchical structure, where the local 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. Here cross-modal interactions are limited to a single segment only (where the input modalities are aligned), covering a short timespan. This does not allow multimodal interactions between non-aligned inputs - and we note that the content of human speech is not always precisely aligned with its corresponding visual contexts in time [52]. In contrast, our method allows global cross-modal interactions, unconstrained by temporal alignment. More recent works [25, 48] explore deeper interactions between video frame features and features from other modalities, by fusing modalities earlier, at the input level itself. In these works, however, inputs from multiple modalities are fed into a single transformer, with 'modality specific' encodings to distinguish between the modalities. In contrast, our two stream transformer decouples within-modality interactions in individual modality streams and allows cross-modality interactions with lateral self-attention blocks. Another differentiater is the fact that all these works operate on scene-level features, while we focus on objects. Thanks to off-the-shelf high-quality object detection [30, 64] and simple bounding box representations, object-centric features have been widely used in a number of image and language problems [4, 51]. In particular, ViLBERT [50] feeds object features and language inputs to a co-attentional transformer. Extending such methods to video, however, is non-trivial, given the number of frames in a video. While object detectors work well on single frames, obtaining reliable tracklet based video features is still an open problem.

## 3. Future Utterance Prediction

We begin by proposing a new future utterance prediction task. The goal is to predict the next utterance in a video, given the previous multimodal context (Figure 1). While next utterance prediction can be evaluated as a generative task [15, 26, 45], we simplify the problem to be selection among pre-collected candidates. Precisely speaking, given a video clip $\mathcal{V} = (F, W)$ where $F = \{f_i\}_{i=1}^{N_f}$ is a sequence of video frames and $W = \{w_i\}_{i=1}^{N_w}$ is a sequence of transcribed words, our goal is to select the true next utterance $u_{\mathcal{T}}$ from a set of candidates $U = \{u_i\}_{i=1}^{M}$ where $\mathcal{T}$ is the index of the true element in $U$ and we set $M = 100$ (Figure 1). Performance is then assessed by ranking the candidates and using popular retrieval metrics.

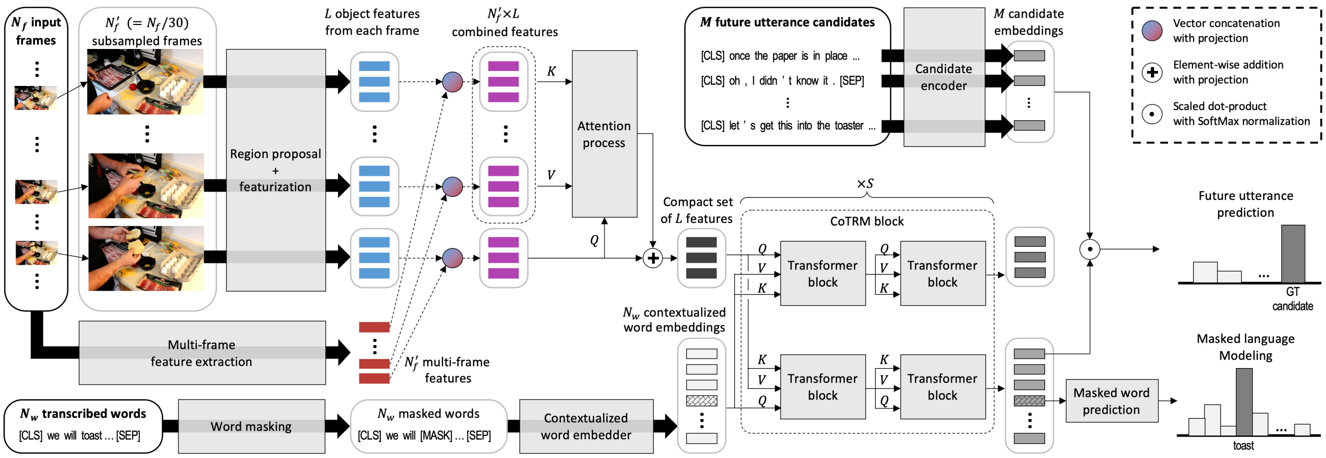



Figure 2: **Co-attentional Multimodal Video Transformer:** Visual depiction of our multimodal training network with $L = 3$ object features per frame. Our model consists of 2 streams, a visual stream which operates on spatiotemporal features, and a text steam which ingests word level features. Our model is trained using two losses, a future utterance prediction ranking loss and a masked language modelling loss.

Our reasons for opting for ranking rather than generation are two fold: (i) Metrics for evaluating language generation (*e.g.* BLEU [62], METEOR [8]) are focused on local matches (n-grams, longest matching sequences, etc). By definition, such metrics are limited to local contexts, and struggle to account for complex sentence structures and word semantics. It is therefore widely accepted that these metrics do not align well with human ratings [68]; and (ii) The output distribution in sentence generation is multimodal, *i.e.* the same information can be paraphrased in multiple ways, leading to many correct answers. There is also inherent ambiguity in the future – given the observation of the present, multiple predictions about the future are possible [23]. In language generation tasks, this is handled by collecting multiple ground truth answers, a strategy which is expensive and difficult to scale. In contrast, popular ranking metrics such as recall@$k$ are better able to assess model performance in tasks with a multimodal output distribution [15, 41].

We note here that unlabelled videos can be used to generate data for our task in a scalable manner. A list of future candidate utterances can be created automatically, with the positive sample being the next utterance in the video, and making the assumption that randomly sampled utterances from different video clips are likely to be negatives [76].

In this work, we use the videos from the HowTo100M dataset [54], as this is a large dataset of 1.2M instructional videos where the speech is usually explicitly linked to the visual content in the video. Examples of future prediction candidates for this dataset can be seen in Figure 3. We use 90% of the videos in HowTo100M for training, and reserve 5% each for validation and test respectively. We name this benchmark How2FUP (more details are provided in Section 5.1).

# 4. Model

Our goal is to effectively learn from both vision and text in a video. We propose a Co-attentional Multimodal Video Transformer (CoMVT), which given a video clip $\mathcal{V}$, extracts contextualized word embeddings and visual features from transcribed words and video frames respectively, and fuses the extracted features to form a multimodal video feature using a co-attentional transformer. We first describe our network architecture, and then the losses used to train this model to solve the task described above (both network and losses are visually depicted in Figure 2).

## 4.1. Network Architecture

### 4.1.1 Text Input Features

Given a sequence of transcribed words[2] $W$, we extract $N_w = |W|$ contextualized word embeddings $e_i$ using BERT [19].

### 4.1.2 Visual Input Features

We extract two types of visual features – multi-frame scene level features, and object level features extracted per frame.

**Scene-level:** We first extract $N_f'$ multi-frame features $m_i$ by feeding $F = \{f_i\}_{i=1}^{N_f}$ frames into S3D [84], a 3D CNN model which has been used in previous approaches in multimodal representation learning [76, 77]. Note that $N_f' (\leq N_f)$ is the number of extracted multi-frame features (determined by the stride and temporal downsampling rate of S3D). Similarly to [77], for every non-overlapping 1-second-long segment of the video, we sample 30 frames and obtain a single feature vector by applying global average-pooling spatiotemporally to the feature activations before the final classifier. This gives us one multi-frame feature $m_i$ per second.

**Object-level:** While multi-frame features are effective at capturing spatiotemporal dynamics, they limit access to individual concepts or objects by squeezing information into a single vector. To overcome this, we also extract object-level features corresponding to single visual objects in each frame. We first

[2] We use WordPiece tokenization from the BERT vocabulary.

subsample $N'_f$ frames building $F' = \{f'_i\}_{i=1}^{N'_f}$ where each frame $f'_i$ is temporally aligned with a multi-frame feature $m_i$. Single-frame object features are then extracted from top-scoring bounding box proposals in each $f'_i$. Following [11], bounding boxes are proposed by a region proposal network (RPN) in [65] and featurized using Graph-Regularized Image Semantic Embeddings (Graph-RISE) [37]. That is, the single-frame object features $\{o_{ij}\}_{j=1}^{L}$ are extracted from each frame $f'_i$ by

$$B_i = \mathrm{RPN}(f'_i) \tag{1}$$

$$o_{ij} = \text{Graph-RISE}(b_{ij}; f'_i), \quad b_{ij} \in B_i \tag{2}$$

where $B_i = \{b_{ij}\}_{j=1}^{L}$ is a set of top $L$ bounding box in $f'_i$ proposed by RPN.

Note that $o_{ij}$ is object-specific but lacks temporal information whereas $m_i$ encodes temporal dynamics without allowing object-specific access. Therefore, we construct combined spatiotemporal visual features that provide both temporal information and object-specific access by merging these two types of features:

$$v_{ij}^{\mathrm{st}} = g_{\mathrm{comb}}([o_{ij}; m_i]) + \mathrm{pos}(o_{ij}) \tag{3}$$

where [;] and $g_{\mathrm{comb}}$ are a vector concatenation operation and a two-layer perceptron, respectively, and $\mathrm{pos}(o_{ij})$ is a positional encoding. The positional encoding $\mathrm{pos}(o_{ij})$ captures the spatiotemporal location of an object $o_{ij}$ and is computed by the sum of sinusoidal encoding of temporal location [81] and linear projection of bounding box coordinates[3].

**Compact feature set extraction:** Since the set of combined visual features $\{v_{ij}^{\mathrm{st}}\}$ is extracted from the entire spatiotemporal space of a video, this leads to a large number of features (often redudant as the same object is detected in multiple frames), which significantly increases complexity of the transformer (see Table 2). Therefore, we construct a more compact set of visual features by aggregating redundant features through an attention process. We select a temporal anchor point $t$ where the object features $v_{tj}^{\mathrm{st}}$ in frame $f'_t$ are used as queries. We then attend to the remaining features as follows: $V_{\mathrm{target}} = \{v_{ij'}^{\mathrm{st}} | i \neq t\}$. This way, the model retrieves features relevant to the query (learned during training). Formally, our model computes a scalar attention score $\alpha_j(v)$ for each element $v \in O_{\mathrm{target}}$ from a $j_{\mathrm{th}}$ query object, and obtains an attended visual feature $v_j^{\mathrm{att}}$ by

$$\alpha_j(v) = \frac{g_{\mathrm{query}}(v_{tj}^{\mathrm{st}}) \cdot g_{\mathrm{key}}(v)}{\sum_{v' \in V_{\mathrm{target}}} g_{\mathrm{query}}(v_{tj}^{\mathrm{st}}) \cdot g_{\mathrm{key}}(v')} \tag{4}$$

$$v_j^{\mathrm{att}} = g_{\mathrm{output}}\left( \sum_{v' \in V_{\mathrm{target}}} \alpha_j(v') g_{\mathrm{value}}(v') \right) \tag{5}$$

where $g_{\mathrm{query}}$, $g_{\mathrm{key}}$, $g_{\mathrm{value}}$ and $g_{\mathrm{output}}$ are all linear projection functions, and $\cdot$ is a scaled dot product proposed in [81]. The final visual feature $v_j^{\mathrm{compact}}$ is then computed from the sum of the anchor object features $v_{tj}^{\mathrm{st}}$ and the attended object features $v_j^{\mathrm{att}}$:

$$v_j^{\mathrm{compact}} = g_{\mathrm{proj}}(v_{tj}^{\mathrm{st}} + v_j^{\mathrm{att}}) \tag{6}$$

[3]We use the normalized coordinates of top-left and bottom-right corners of bounding boxes.

where $g_{\mathrm{proj}}$ is a projection function implemented by a two-layer perceptron.

#### 4.1.3 Co-attentional Transformers

Features from text $w_i$ and vision $v_j^{\mathrm{compact}}$ are then fused using an architecture similar to the co-attentional transformer (CoTRM) proposed in [50]. A CoTRM block is composed of four transformer (TRM) blocks that compute attention distributions over values by computing a scaled dot product between queries and keys, and obtaining a set of output vectors from weight-averaged values [81].

A CoTRM block consists of two streams, each built by stacking two TRM blocks. The first TRM block in each stream takes two multimodal inputs sets: one for queries and the other for keys and values alternating their roles in each stream. The second TRM block is independant within a modality stream.

Formally, given two sets of input features $V^{(s)}$ and $E^{(s)}$, the visual features in $V^{(s)}$ at $s_{\mathrm{th}}$ CoTRM block are contextualized by

$$\hat{V}^{(s)} = \mathrm{TRM}(V^{(s)}, E^{(s)}) \tag{7}$$

$$V^{(s+1)} = \mathrm{TRM}(\hat{V}^{(s)}, \hat{V}^{(s)}) \tag{8}$$

where $\mathrm{TRM}(Q, K)$ is a TRM block with query inputs $Q$ and key-value inputs $K$. Note that the first TRM block, *i.e.* equation (7), performs inter-modality contextualization by adding related word features to each visual feature whereas the second block, *i.e.* equation (8), performs intra-modality contextualization through the regular self-attention mechanism. Similarly, the word embeddings $E^{(s)}$ are contextualized by

$$\hat{E}^{(s)} = \mathrm{TRM}(E^{(s)}, V^{(s)}) \tag{9}$$

$$E^{(s+1)} = \mathrm{TRM}(\hat{E}^{(s)}, \hat{E}^{(s)}). \tag{10}$$

We repeat this process $S$ times and set the initial inputs $V^{(0)} = \{v_i^{\mathrm{compact}}\}_{i=1}^{L}$ and $E^{(0)} = \{e_i\}_{i=1}^{N_w}$.

The two stream nature of CoTRMs inherently treats each modality separately allowing modality-specific operations and representations through different parameterizations of TRMs in the streams.

### 4.2. Training Objectives

We train our model with the following two losses:

**1) Next Utterance Prediction Loss:** Here we treat the textual modality as the main modality and treat $e_1^{(S)}$ as the embedding of all multimodal inputs. Note that $e_1^{(S)}$ corresponds to the contextualized embedding of the special '[CLS]' token added to the input. Since our goal is to choose the true next utterance $u_{\mathcal{T}}$ from a set of candidate utterances $U = \{u_i\}_{i=1}^{M}$, we first embed each candidate utterance using an additional BERT encoder and predict the probability of $u_i$ being the true next utterance $P(u_i | e_1^{(S)}, U)$ by

$$P(u_i | e_1^{(S)}, U) = \frac{\exp\left(e_1^{(S)} \cdot g_{\mathrm{cand}}(u_i)\right)}{\sum_{u' \in U} \exp\left(e_1^{(S)} \cdot g_{\mathrm{cand}}(u')\right)} \tag{11}$$

where $g_{\mathrm{cand}}$ is the candidate embedding function for which we use a BERT encoder. Note that this candidate utterance encoder is also fine-tuned during training and has different parameters from those in the input text encoder. We train the network by minimizing the negative log-likelihood of the true next utterance $-\log P(u_{\mathcal{T}}|e_1^{(S)}, U)$ where $U$ is constructed by collecting all ground-truth next utterances of examples within each batch during training.

**2) Masked Language Modelling:** In addition to our next utterance prediction loss, we implement the masking scheme and loss function introduced in [19] and mask out some of input words. We apply this loss on downstream evaluations as well. This appears to have a regularisation effect, similar to dropout [75]. We additionally explored visual input masking as in [76], but found little changes in performance.

**Implementation details.** We use the BERT base model for the contextualized word embedding extraction and test the proposed networks with $S \in \{1,2,4\}$. RPN [65], GRAPH-Rise [37] and S3D [84] are initialized and fixed with pretrained weights; all the other parameters are updated during training in all experiments. We set the maximum lengths for the transcribed words $N_w$ and the downsampled video frames $N'_f$ to 128 and 30, respectively, and truncate longer sequences keeping the last elements.

# 5. Experiments

We first train our model for Future Utterance Prediction and show results on two datasets, HowToFUP and Coin-FUP. We then take the model pretrained on this task and demonstrate that it generalises well to VideoQA datasets, achieving state-of-the-art results. The input/output configurations for these tasks are described in Appendix B. The next section describes all the datasets used in this work, and then delves into experimental details.

## 5.1. Datasets

### 5.1.1 Future Utterance Prediction

**HowToFUP:** We repurpose HowTo100M [54], a large-scale dataset of 1.2M instructional videos for the task of Future Utterance Prediction. Transcripts are obtained using the YouTube ASR API [1], however these are noisy (Figure B in the Appendix shows an example). Videos that have been taken down from YouTube are not used. We then divide these videos into shorter segments, henceforth referred to as video clips. The duration of video clips is determined as follows: we start with a single ASR sentence and then iteratively expand the length of the video clip backwards by adding previous sentences until the segment is longer than 5 seconds. Each video clip therefore contains full sentences in the ASR (no sentences are cut-off mid way). This process results in 35M training examples and 2M examples each in the validation and test splits. In order to create diverse validation and test sets, we then further reduce the number of clips in each by randomly subsampling 6% of the clips (as many clips contain redundant input contexts). The final validation and test splits consist of 120K clips, and are used for testing all models.

For each video clip, we then create a list of $M=100$ future utterance candidates through random sampling. $M-1$ negative candidates are sampled from the entire answer pool to build $U$ for the test and validation splits.

We note that this dataset is an order of magnitude larger in number of datapoints than existing video captioning datasets, as well as Conceptual Captions [73], the largest publicly released image captioning dataset widely used for pretraining vision-text models in the image domain [12, 50], however is noisier due to (i) errors caused by imperfect ASR and (ii) given the ASR is generated from continuous narration, it often consists of incomplete sentences that lack punctuation. A further analysis in provided in Appendix D.

**COIN-FUP:** We also repurpose COIN [79], another dataset of instructional videos to evalute the task of future utterance prediction. This dataset is smaller, with 12K videos. We follow the same clip generation pipeline used for HowToFUP[4] to create COIN-FUP, resulting in 78K, 5K and 4K examples in the train, validation and test splits respectively.

### 5.1.2 Next Step Prediction

**COIN-NSP:** Unlike HowTo100M, COIN [79] also contains additional manually annotated categorical steps labelled for each video. Hence we also investigate the performance of a related, albeit slightly different task on this dataset – next step prediction (NSP). Unlike FUP, where the goal is to select from a list of utterances in free form natural language, NSP focuses on predicting the next step from a list of pre-defined action categories. Similarly to FUP example generation, we automatically construct COIN-NSP by iterating over each step annotation and extract its precedent multimodal video segment generating 18K training examples and 1K examples for both validation and test splits with 735 step classes (24.5 examples per class on average in the entire dataset). Note that COIN-NSP is smaller than COIN-FUP in the number of examples since there are fewer manual step annotations than total number of utterances. At inference time, we simply replace the candidate selection component in our model with a softmax classifier.

**CrossTask-NSP:** CrossTask [90] is another dataset that contains manual annotations of steps for instructional videos of 18 pre-selected tasks. We create CrossTask-NSP following the COIN-NSP construction process resulting in 14K training examples and 2K validation/test examples with 105 target classes.

### 5.1.3 Downstream VideoQA Benchmarks

**MSRVTT-QA and MSVD-QA:** MSRVTT-QA and MSVD-QA are popular video question answering benchmarks introduced in [85]. We use publicly available features and follow the standard train, val and test splits used in [85]: 158K, 12K and 73K QA pairs for MSRVTT-QA and 31K, 6K and 13K pairs for MSVD-QA.

**ActivityNet-QA:** ActivityNet-QA [88] contains 58K open-ended QA annotations where the train, val and test splits have 32K, 18K and 8K QA pairs, respectively.

**How2QA:** How2QA [47] consists of QA annotations for the HowTo100M dataset. It contains 35K train and 3K publicly available val samples. Each question has three negative answers and one correct answer.

[4] However we do not subsample in the validation and test sets, to maintain a reliable number of samples for evaluation.

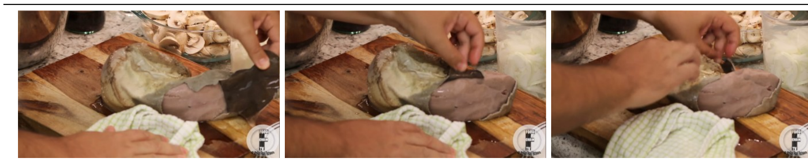
| Inputs (video frames and utterances) | Prediction (future utterance) | |
|---|---|---|
| <br>**Transcript**: It was completely rotten on the inside rotten. Yeah, because I had you waited till a because I thought it's gonna keep getting bigger, but it was just because I didn't have a green thumb. | *It's grown.*<br>*Now do we have a finished product?*<br>*That was my **watermelon** had completely ripened weeks ago.*<br>*I don't think I would drive a bit long enough to get in there.*<br>... | ✓ (*That was my **watermelon** had completely ripened weeks ago.*) |
| <br>**Transcript**: And here's another one that might be close, but I just don't think they're quite ready yet, but it's always nice to check so they need a little bit more growth time. All right. | *We like to eat with bone like quit bone.*<br>*You need to see what these things are.*<br>*So here are the **carrots**.*<br>*They're looking fabulous.*<br>... | ✓ (*So here are the **carrots**.*) |
| <br>**Transcript**: They just shake it up and pretty much the exact same thing as spraying you all should be familiar with heavy scraping. | *That's bad.*<br>*But so basically I just spray the **folder** and then I just paste this down and I try to get them in all the exact same position.*<br>*It doesn't take much time at all.*<br>*We need to mount this on our a/c condenser.*<br>... | ✓ (*But so basically I just spray the **folder** and then I just paste this down and I try to get them in all the exact same position.*) |
| <br>**Transcript**: This skin should come right off just like that. Okay, and there we go. | *So underneath that is some really really nice **meat**.*<br>*It is nice to write down a date on the jar.*<br>*I guess the skin anyway, so it's ready.*<br>*Don't click it.*<br>... | ✓ (*So underneath that is some really really nice **meat**.*) |

Figure 3: **Qualitative results on HowToFUP.** On the right, we show the results of the baseline model that uses text inputs only (highlighted in red) and our multimodal model (highlighted in green). The GT utterance has a ✓ next to it. Note how the transcript often contains phrases with subtle indications to visual content, such as 'here's another one' (second row) and 'should come off right like that' (fourth row). In many of these cases, the correct future utterance refers to an object which can only be known from the visual context (highlighted in bold). The text only model often selects generics utterances, or those which are referred to specifically in previous dialogue (fourth row, selected candidate has the word 'skin'). Further examples are provided in Appendix C.

### 5.2. Baselines

We compare our model to a number of single and multiple modality baselines.

**S3D - visual only:** We use the S3D [84] model pretrained on Kinetics applied to video frames.

**Text only Baseline:** For the text only baseline, we use BERT [19], which is the winning model in the Eighth Dialog System Technology Challenge for response prediction in text only dialog benchmarks [29].

**Single-Stream Multimodal Baseline:** We also implement a single stream transformer operating on a single multimodal input stream, which is the most widely used framework for video encoding with multimodal inputs [25, 42, 48, 77]. We adopt the architecture used in [48] and train the network using the same next utterance prediction loss for FUP. Note that this architecture is slightly different from that of BERT, and hence we cannot use pre-trained BERT weights.

In addition to our full model, we also show results without the object level features referred to as 'CoMVT (Scene feats only)' in Table 1, as this is more similar to previous multimodal models [25, 42, 48, 77], as well as show the effects without BERT pretraining for the text stream and the MLM loss. For each model including the baselines, we perform grid search on learning rates and report the test performance of the best models in the validation set. On HowToFUP, every network is trained for 2M with a batch size of 512. The learning rate is warmed up for 10K iterations and is continuously decayed per every 30K iterations by the factor of 0.95. On the other datasets, due to the small sizes of the datasets, the models are trained for 20K iterations with a 50 iteration warm-up period and 1K decay length.

### 5.3. Results

#### 5.3.1 Future Utterance Prediction

Table 1 shows the recall at $k \in \{1,5\}$ (R@$k$) on HowToFUP. All multimodal models outperform text only baselines showing the value of visual inputs for this task. Our best model results in an

Table 1: Recall at $k \in \{1,5\}$ on HowToFUP. **BERT PT:** Input text and candidate encoders initialised using BERT pretrained weights. **MLM:** Masked Language Modelling loss. †Multimodal Single-stream baseline used in a number of works [48, 76, 77]. $S=2$ by default for all rows except the last one where $S=4$.

| Model | BERT PT | MLM | $S=4$ | R@1 | R@5 |
|---|---|---|---|---|---|
| S3D (Vision only) | | | | 5.85 | 18.24 |
| Baseline (Text only) | | | | 59.21 | 81.31 |
| | ✓ | | | 60.90 | 82.59 |
| Multimodal Baseline† | | | | 62.73 | 84.78 |
| CoMVT (Scene feats only) | | | | 63.35 | 85.22 |
| | ✓ | | | 65.64 | 86.67 |
| | ✓ | ✓ | | 67.13 | 87.42 |
| CoMVT (Combined feats) | ✓ | | | 66.82 | 87.38 |
| | ✓ | ✓ | | 67.79 | 88.00 |
| | ✓ | ✓ | ✓ | **68.34** | **88.28** |

Table 2: Flops and R@$k$s with and without compact feature set extraction on HowToFUP. All models trained with MLM. Note we cannot train with $S=4$ without compact feature set extraction due to memory constraints.

| Model | $S$ | Gflops | R@1 | R@5 |
|---|---|---|---|---|
| w/o compact feature set extraction | 2 | 8.5 | 67.83 | 88.05 |
| | 4 | 12.7 | N/A | N/A |
| w/ compact feature set extraction | 2 | 6.8 (-20.2%) | 67.79 | 88.00 |
| | 4 | 7.7 (-39.3%) | 68.34 | 88.28 |

8% improvement in R@1. The gain due to visual input is also demonstrated by the examples in Figure 3 and Appendix C.

We next ablate various aspects of our model and training setup.

**Architecture Components:** Table 1 shows the incremental value of different components in our architecture. Our two stream model surpasses the single stream multimodal baseline model while using the same scene-level features. It is also interesting that both BERT pretraining and the masked language modeling loss improve performance even though we train on a large-scale dataset with more than 35M training examples. Finally, the use of the combined features in our full model and additional CoTRM blocks show additional gains.

**Efficiency:** We also analyze the efficiency gains of our compact feature set extraction module (see Table 2 for flops[5] and R@$k$ results). Using our compact feature set module significantly reduces flops by 20.2% and 39.3% with $S=2$ and 4, respectively, while maintaining performance with $S=2$. For $S=4$, we are unable to obtain R@$k$s without compact feature set extraction on our TPU configurations due to significant memory consumption during training.

**Effect of Training Data:** We also perform an ablation study analysing the effect of training data size on performance. We train on different fractions of the HowToFUP training set (results in Figure 4), and show steep performance drops when the size of the training set is reduced. We also note that the trend points towards

[5]Flops are measured per sample by profiling evaluation steps on TPUs.

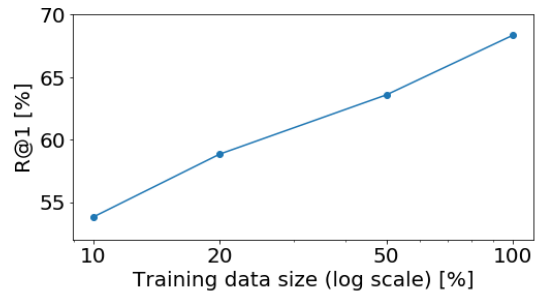



Figure 4: Effects of training data size on HowToFUP performance, reported as R@1.

Table 3: Results on COIN-FUP. $S=1$ by default. **HowTo PT**: Entire model is pretrained on HowToFUP with $S=4$.

| Model | BERT PT | MLM | R@1 | R@5 |
|---|---|---|---|---|
| S3D (Vision only) | | | 2.97 | 10.44 |
| Baseline (Text only) | | | 19.82 | 43.19 |
| | ✓ | | 35.68 | 64.42 |
| Multimodal Baseline | | | 21.02 | 45.82 |
| CoMVT | | | 22.40 | 47.25 |
| | ✓ | | 37.08 | 66.90 |
| | ✓ | ✓ | 39.11 | 68.22 |
| CoMVT (HowTo PT) | | ✓ | **70.92** | **93.52** |

linear improvements in R@1 as the training set size is doubled. Given that the performance does not seem to be saturated yet, we hypothesise that further performance gains are possible by scaling up with instructional video data beyond HowTo100M.

Future Utterance Prediction results for COIN are shown in Table 3. Similar trends hold, however we also note that pretraining on HowToFUP provides a massive boost in performance (over 30% value increase in R@1). This significant gain can be explained by the relatively small size of COIN-FUP.

#### 5.3.2 Next Step Prediction

The results for Next Step Prediction (a classification task) on COIN-NSP and CrossTask-NSP are provided in Table 4. Interestingly, using visual inputs shows more of an improvement over the text only baseline for this task compared to FUP. Note that NSP contains more examples of humans activities (while FUP has a more diverse set of utterances). In addition to the baselines described in Section 5.2, we also compare to two state-of-the-art vision-only models for next step prediction, RULSTM [24] and TAR [66][6]

Pretraining on BERT and using the MLM loss particularly help prevent our multimodal model from overfitting on this small dataset. The largest gain comes from pretraining on HowToFUP, which also allows us to increase the value of $S$ in the model without suffering from overfitting. We also show results of our model

[6]We reimplemented these models to use our extracted features to provide a fair comparison.

| Inputs (video frames and utterances) | Prediction (next step class) |
|---|---|
|  | *install new door knob* ✓<br>*boil water or coffee*<br>*install bulb and light housing*<br>*close switch*<br>*drive car forward* |
| **Transcript**: The process is simply reversed insert the new dead light into the hole at the end of the door. | ··· 730 more step classes ··· |

Figure 5: **Qualitative results on COIN-NSP.** On the right, we show the predicted class (out of 735 classes) by the text only baseline ( red ) and the correct class identified by our multimodal model ( green ). Note that an ASR error (highlighted by underline, 'dead light' should be 'deadlatch') causes the text only baseline to make a mistake (prediction about light), which is corrected by access to visual inputs in our model.

Table 4: Results on COIN-NSP and CrossTask-NSP. *fixed dummy text is fed to utilize visual inputs only.

| Model | HowTo PT | BERT PT | MLM | COIN R@1 | COIN R@5 | CrossTask R@1 | CrossTask R@5 |
|---|---|---|---|---|---|---|---|
| S3D (Vision only) | | | | 29.20 | 69.99 | 29.34 | 72.38 |
| RULSTM (Vision only) | | | | 26.38 | 56.70 | 28.39 | 67.90 |
| TAR (Vision only) | | | | 18.13 | 33.64 | 17.64 | 46.03 |
| Baseline (Text only) | | | | 16.31 | 42.80 | 17.77 | 50.59 |
| | | ✓ | | 20.54 | 47.13 | 20.29 | 58.83 |
| Multimodal Baseline | | | | 24.97 | 65.26 | 30.73 | 74.09 |
| CoMVT | | | | 28.20 | 67.37 | 30.66 | 70.51 |
| | | ✓ | | 30.31 | 73.01 | 31.77 | 75.59 |
| | | ✓ | ✓ | 33.33 | 74.52 | 33.73 | 78.21 |
| | ✓ | | ✓ | **37.46** | **76.64** | **42.12** | **81.45** |
| CoMVT (Vision only*) | ✓ | | ✓ | 33.84 | 68.78 | 39.19 | 81.26 |

Table 5: Comparison to state-of-the-art on MSVD-QA and MSRVTT-QA. We report top 1 accuracy [%]. We show results of our model with and without pretraining on HowToFUP.

| Methods | MSVD-QA | MSRVTT-QA |
|---|---|---|
| ST-VQA [35] | 31.3 | 30.9 |
| Co-Mem [27] | 31.7 | 32.0 |
| AMU [85] | 32.0 | 32.5 |
| HMEMA [20] | 33.7 | 33.0 |
| HRA [13] | 34.4 | 35.1 |
| SSML [3] | 35.1 | 35.1 |
| HCRN [43] | 36.1 | 35.6 |
| CoMVT (scratch) | 35.7 | **37.3** |
| CoMVT (pretrained) | **42.6** | **39.5** |

using visual inputs only (with pretrained weights), and find that we obtain decent results (this is achieved by feeding in a dummy text input with an empty sentence, *i.e.*, a sequence containing a [CLS] token and a [SEP] token). A qualitative result is shown in Figure 5.

Table 6: Comparison to state-of-the-art on ActivityNet-QA.

| Model | Accuracy |
|---|---|
| E-VQA [88] | 25.10 |
| E-MN [88] | 27.10 |
| E-SA [88] | 31.80 |
| MAR-VQA [89] | 34.60 |
| CoMVT (scratch) | **36.63** |
| CoMVT (pretrained) | **38.75** |

Table 7: Comparison to state-of-the-art on the How2QA validation set.

| Model | Accuracy |
|---|---|
| HERO [47] | 74.10 |
| CoMVT (scratch) | **78.04** |
| CoMVT (pretrained) | **82.29** |

#### 5.3.3 Transfer Learning to Video QA

We additionally show results of our model pretrained on HowToFUP and then fine-tuned on 4 popular video QA benchmarks, in Table 5 for MSRVTT-QA and MSVD-QA, Table 6 for ActivityNet-QA, and Table 7 for How2QA. For all datasets, our network architecture trained from scratch already outperforms or performs comparably to the existing state of the art; and finetuning from pretrained weights on HowToFUP provides a further boost. We note that our model is pretrained without any QA supervision at all, and generalises well to video QA.

## 6. Conclusion

We propose a new visually conditioned Future Utterance Prediction (FUP) learning task, where the goal is to predict the next utterance in an instructional video using both visual frames and transcribed speech. We set benchmarks on both the HowTo100M and COIN datasets, and show state-of-the-art results on downstream video QA benchmarks. We hope that this work will increase interest in the exciting field of visually contextualized dialogue systems.

## References


[1] YouTube Data API. https://developers.google.com/youtube/v3/docs/captions. 5

[2] Huda Alamri, Vincent Cartillier, Abhishek Das, Jue Wang, Anoop Cherian, Irfan Essa, Dhruv Batra, Tim K Marks, Chiori Hori, Peter Anderson, et al. Audio visual scene-aware dialog. In *CVPR*, 2019. 2

[3] Elad Amrani, Rami Ben-Ari, Daniel Rotman, and Alex Bronstein. Noise estimation using density estimation for self-supervised multimodal learning. *arXiv preprint arXiv:2003.03186*, 2020. 8

[4] 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 *CVPR*, 2018. 2

[5] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *ICCV*, 2015. 2

[6] Layla El Asri, Hannes Schulz, Shikhar Sharma, Jeremie Zumer, Justin Harris, Emery Fine, Rahul Mehrotra, and Kaheer Suleman. Frames: A corpus for adding memory to goal-oriented dialogue systems. *arXiv preprint arXiv:1704.00057*, 2017. 1

[7] Max Bain, Arsha Nagrani, Andrew Brown, and Andrew Zisserman. Condensed movies: Story based retrieval with contextual embeddings. *ACCV*, 2020. 2

[8] Satanjeev Banerjee and Alon Lavie. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, 2005. 3

[9] Olfa Ben-Ahmed and Benoit Huet. Deep multimodal features for movie genre and interestingness prediction. In *International Conference on Content-Based Multimedia Indexing (CBMI)*, 2018. 2

[10] Paweł Budzianowski, Tsung-Hsien Wen, Bo-Hsiang Tseng, Inigo Casanueva, Stefan Ultes, Osman Ramadan, and Milica Gašić. Multiwoz-a large-scale multi-domain wizard-of-oz dataset for task-oriented dialogue modelling. *arXiv preprint arXiv:1810.00278*, 2018. 1

[11] Soravit Changpinyo, Bo Pang, Piyush Sharma, and Radu Soricut. Decoupled box proposal and featurization with ultrafine-grained semantic labels improve image captioning and visual question answering. In *EMNLP-IJCNLP*, 2019. 4

[12] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Learning universal image-text representations. *arXiv preprint arXiv:1909.11740*, 2019. 5

[13] Muhammad Iqbal Hasan Chowdhury, Kien Nguyen, Sridha Sridharan, and Clinton Fookes. Hierarchical relational attention for video question answering. In *ICIP*, 2018. 8

[14] Yun-Wei Chu, Kuan-Yen Lin, Chao-Chun Hsu, and Lun-Wei Ku. Multi-step joint-modality attention network for scene-aware dialogue system. *arXiv preprint arXiv:2001.06206*, 2020. 2

[15] Abhishek Das, Satwik Kottur, Khushi Gupta, Avi Singh, Deshraj Yadav, José MF Moura, Devi Parikh, and Dhruv Batra. Visual dialog. In *CVPR*, 2017. 1, 2, 3

[16] Abhishek Das, Satwik Kottur, José MF Moura, Stefan Lee, and Dhruv Batra. Learning cooperative visual dialog agents with deep reinforcement learning. In *ICCV*, 2017. 2

[17] Kees van Deemter, Mariët Theune, and Emiel Krahmer. Real versus template-based natural language generation: A false opposition? *Computational linguistics*, 31(1):15–24, 2005. 2

[18] Chaorui Deng, Qi Wu, Qingyao Wu, Fuyuan Hu, Fan Lyu, and Mingkui Tan. Visual grounding via accumulated attention. In *CVPR*, 2018. 2

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 2, 3, 5, 6

[20] Chenyou Fan, Xiaofan Zhang, Shu Zhang, Wensheng Wang, Chi Zhang, and Heng Huang. Heterogeneous memory enhanced multimodal attention model for video question answering. In *CVPR*, 2019. 8

[21] David F Fouhey, Wei-cheng Kuo, Alexei A Efros, and Jitendra Malik. From lifestyle vlogs to everyday interactions. In *CVPR*, 2018. 2

[22] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *EMNLP*, 2016. 2

[23] Antonino Furnari, Sebastiano Battiato, and Giovanni Maria Farinella. Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation. In *ECCV*, 2018. 3

[24] Antonino Furnari and Giovanni Farinella. Rolling-unrolling lstms for action anticipation from first-person video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 7

[25] Valentin Gabeur, Chen Sun, Karteek Alahari, and Cordelia Schmid. Multi-modal transformer for video retrieval. In *ECCV*, 2020. 2, 6

[26] Jianfeng Gao, Michel Galley, and Lihong Li. Neural approaches to conversational ai. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018. 2

[27] Jiyang Gao, Runzhou Ge, Kan Chen, and Ram Nevatia. Motion-appearance co-memory networks for video question answering. In *CVPR*, 2018. 8

[28] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in VQA matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 2

[29] Chulaka Gunasekara, Jonathan K. Kummerfeld, Luis Lastras, and Walter S. Lasecki. Noesis ii: Predicting responses, identifying success, and managing complexity in task-oriented dialogue. In *8th Edition of the Dialog System Technology Challenges at AAAI 2020*, 2020. 6

[30] Chuchu Han, Jiacheng Ye, Yunshan Zhong, Xin Tan, Chi Zhang, Changxin Gao, and Nong Sang. Re-id driven localization refinement for person search. In *ICCV*, 2019. 2

[31] Matthew Henderson, Blaise Thomson, and Jason D Williams. The second dialog state tracking challenge. In *SIGDIAL*, 2014. 1

[32] Chiori Hori, Huda Alamri, Jue Wang, Gordon Wichern, Takaaki Hori, Anoop Cherian, Tim K Marks, Vincent Cartillier, Raphael Gontijo Lopes, Abhishek Das, et al. End-to-end audio visual scene-aware dialog using multimodal attention-based video features. In *ICASSP*, 2019. 2

[33] Chiori Hori, Anoop Cherian, Tim K Marks, and Florian Metze. Audio visual scene-aware dialog track in dstc8. 2

[34] Oana Ignat, Laura Burdick, Jia Deng, and Rada Mihalcea. Identifying visible actions in lifestyle vlogs. *arXiv preprint arXiv:1906.04236*, 2019. 2

[35] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Tgif-qa: Toward spatio-temporal reasoning in visual question answering. In *CVPR*, 2017. 8

[36] Weike Jin, Zhou Zhao, Mao Gu, Jun Xiao, Furu Wei, and Yueting Zhuang. Video dialog via progressive inference and cross-transformer. In *EMNLP-IJCNLP*, 2019. 2

[37] Da-Cheng Juan, Chun-Ta Lu, Zhen Li, Futang Peng, Aleksei Timofeev, Yi-Ting Chen, Yaxi Gao, Tom Duerig, Andrew Tomkins, and Sujith Ravi. Graph-rise: Graph-regularized image semantic embedding. *arXiv preprint arXiv:1902.10814*, 2019. 4, 5

[38] Seokhwan Kim, Michel Galley, Chulaka Gunasekara, Sungjin Lee, Adam Atkinson, Baolin Peng, Hannes Schulz, Jianfeng Gao, Jinchao Li, Mahmoud Adada, et al. The eighth dialog system technology challenge. *arXiv preprint arXiv:1911.06394*, 2019. 1

[39] Seok-Hwan Kim, Cheong-Jae Lee, Sang-Keun Jung, and GaryGeun-Bae Lee. An example-based natural language dialogue system for epg information access. *Journal of KIISE: Software and Applications*, 34(2):123–130, 2007. 2

[40] Satwik Kottur, José MF Moura, Devi Parikh, Dhruv Batra, and Marcus Rohrbach. Visual coreference resolution in visual dialog using neural module networks. In *ECCV*, 2018. 2

[41] Jonathan K Kummerfeld, Sai R Gouravajhala, Joseph J Peper, Vignesh Athreya, Chulaka Gunasekara, Jatin Ganhotra, Siva Sankalp Patel, Lazaros C Polymenakos, and Walter Lasecki. A large-scale corpus for conversation disentanglement. In *ACL*, 2019. 3

[42] Hung Le and Nancy F Chen. Multimodal transformer with pointer network for the dstc8 avsd challenge. *arXiv preprint arXiv:2002.10695*, 2020. 2, 6

[43] Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. Hierarchical conditional relation networks for video question answering. In *CVPR*, 2020. 8

[44] Cheongjae Lee, Sangkeun Jung, Jihyun Eun, Minwoo Jeong, and Gary Geunbae Lee. A situation-based dialogue management using dialogue examples. In *ICASSP*, 2006. 2

[45] Hwanhee Lee, Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. Dstc8-avsd: Multimodal semantic transformer network with retrieval style word generator. *arXiv preprint arXiv:2004.08299*, 2020. 2

[46] Jiwei Li, Michel Galley, Chris Brockett, Georgios P Spithourakis, Jianfeng Gao, and Bill Dolan. A persona-based neural conversation model. *arXiv preprint arXiv:1603.06155*, 2016. 1

[47] Linjie Li, Yen-Chun Chen, Yu Cheng, Zhe Gan, Licheng Yu, and Jingjing Liu. Hero: Hierarchical encoder for video+ language omni-representation pre-training. In *EMNLP*, 2020. 2, 5, 8

[48] Zekang Li, Zongjia Li, Jinchao Zhang, Yang Feng, Cheng Niu, and Jie Zhou. Bridging text and video: A universal multimodal transformer for video-audio scene-aware dialog. *arXiv preprint arXiv:2002.00163*, 2020. 2, 6, 7

[49] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *BMVC*, 2019. 2

[50] Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. ViLBERT: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In *NeurIPS*, 2019. 2, 4, 5

[51] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Neural baby talk. In *CVPR*, 2018. 2

[52] Antoine Miech, Jean-Baptiste Alayrac, Lucas Smaira, Ivan Laptev, Josef Sivic, and Andrew Zisserman. End-to-end learning of visual representations from uncurated instructional videos. In *CVPR*, 2020. 2

[53] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018. 2

[54] Antoine Miech, Dimitri Zhukov, Jean-Baptiste Alayrac, Makarand Tapaswi, Ivan Laptev, and Josef Sivic. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. In *ICCV*, 2019. 2, 3, 5

[55] Nikola Mrkšić, Diarmuid O Séaghdha, Blaise Thomson, Milica Gašić, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. Multi-domain dialog state tracking using recurrent neural networks. *arXiv preprint arXiv:1506.07190*, 2015. 1

[56] Jonghwan Mun, Minsu Cho, and Bohyung Han. Text-guided attention model for image captioning. In *AAAI*, 2017. 2

[57] Jonghwan Mun, Minsu Cho, and Bohyung Han. Local-global video-text interactions for temporal grounding. In *CVPR*, 2020. 2

[58] Jonghwan Mun, Paul Hongsuck Seo, Ilchae Jung, and Bohyung Han. MarioQA: Answering questions by watching gameplay videos. In *ICCV*, 2017. 2

[59] Jonghwan Mun, Linjie Yang, Zhou Ren, Ning Xu, and Bohyung Han. Streamlined dense video captioning. In *CVPR*, 2019. 2

[60] Arsha Nagrani, Chen Sun, David Ross, Rahul Sukthankar, Cordelia Schmid, and Andrew Zisserman. Speech2action: Cross-modal supervision for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10317–10326, 2020. 2

[61] Hyeonwoo Noh, Paul Hongsuck Seo, and Bohyung Han. Image question answering using convolutional neural network with dynamic parameter prediction. In *CVPR*, 2016. 2

[62] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL*, 2002. 3

[63] Mandela Patrick, Po-Yao Huang, Yuki Asano, Florian Metze, Alexander Hauptmann, João Henriques, and Andrea Vedaldi. Support-set bottlenecks for video-text representation learning. *arXiv preprint arXiv:2010.02824*, 2020. 2

[64] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 2

[65] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 4, 5

[66] Fadime Sener, Dipika Singhania, and Angela Yao. Temporal aggregate representations for long term video understanding. In *ECCV*, 2020. 7

[67] Paul Hongsuck Seo, Andreas Lehrmann, Bohyung Han, and Leonid Sigal. Visual reference resolution using attention memory for visual dialog. In *NIPS*, 2017. 2

[68] Paul Hongsuck Seo, Piyush Sharma, Tomer Levinboim, Bohyung Han, , and Radu Soricut. Reinforcing an image caption generator using off-line human feedback. In *AAAI*, 2020. 2, 3

[69] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. *arXiv preprint arXiv:1507.04808*, 2015. 1

[70] Iulian V Serban, Alessandro Sordoni, Yoshua Bengio, Aaron Courville, and Joelle Pineau. Building end-to-end dialogue systems using generative hierarchical neural network models. In *AAAI*, 2016. 2

[71] Pararth Shah, Dilek Hakkani-Tur, Bing Liu, and Gokhan Tur. Bootstrapping a neural conversational agent with dialogue self-play, crowdsourcing and on-line reinforcement learning. In *NAACL-HLT*, 2018. 1

[72] Yuanlong Shao, Stephan Gouws, Denny Britz, Anna Goldie, Brian Strope, and Ray Kurzweil. Generating high-quality and informative conversation responses with sequence-to-sequence models. In *EMNLP*, 2017. 2

[73] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 2, 5

[74] Vivek Sharma, Makarand Tapaswi, and Rainer Stiefelhagen. Deep multimodal feature encoding for video ordering. *arXiv preprint arXiv:2004.02205*, 2020. 2

[75] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 5

[76] Chen Sun, Fabien Baradel, Kevin Murphy, and Cordelia Schmid. Learning video representations using contrastive bidirectional transformer. *arXiv preprint arXiv:1906.05743*, 2019. 2, 3, 5, 7

[77] Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. In *ICCV*, 2019. 2, 3, 6, 7

[78] Zhongkai Sun, Prathusha K. Sarma, William A. Sethares, and Yingyu Liang. Learning relationships between text, audio, and video via deep canonical correlation for multimodal language analysis. In *AAAI*, 2020. 2

[79] Yansong Tang, Dajun Ding, Yongming Rao, Yu Zheng, Danyang Zhang, Lili Zhao, Jiwen Lu, and Jie Zhou. COIN: A large-scale dataset for comprehensive instructional video analysis. In *CVPR*, 2019. 2, 5

[80] Sebastian Varges and Chris Mellish. Instance-based natural language generation. In *NAACL*, 2001. 2

[81] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017. 4

[82] Jason D Williams. A belief tracking challenge task for spoken dialog systems. In *NAACL-HLT Workshop on Future directions and needs in the Spoken Dialog Community: Tools and Data (SDCTD 2012)*, 2012. 1

[83] Fanyi Xiao, Leonid Sigal, and Yong Jae Lee. Weakly-supervised visual grounding of phrases with linguistic structures. In *CVPR*, 2017. 2

[84] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *ECCV*, 2018. 3, 5, 6

[85] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answering via gradually refined attention over appearance and motion. In *ACM MM*, 2017. 2, 5, 8

[86] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, 2015. 2

[87] Rui Yan. "chitty-chitty-chat bot": Deep learning for conversational ai. In *IJCAI*, volume 18, 2018. 1

[88] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, 2019. 2, 5, 8

[89] Yueting Zhuang, Dejing Xu, Xin Yan, Wenzhuo Cheng, Zhou Zhao, Shiliang Pu, and Jun Xiao. Multichannel attention refinement for video question answering. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 16(1s):1–23, 2020. 8

[90] Dimitri Zhukov, Jean-Baptiste Alayrac, Ramazan Gokberk Cinbis, David Fouhey, Ivan Laptev, and Josef Sivic. Cross-task weakly supervised learning from instructional videos. In *CVPR*, 2019. 5