

Video Question Answering with Phrases via Semantic Roles

Abstract

Video Question Answering (VidQA) evaluation metrics have been limited to a single-word answer or selecting a phrase from a fixed set of phrases. These metrics limit the VidQA models’ application scenario. In this work, we leverage semantic roles derived from video descriptions to mask out certain phrases, to introduce VidQAP which poses VidQA as a fill-in-the-phrase task. To enable evaluation of answer phrases, we compute the relative improvement of the predicted answer compared to an empty string. To reduce the influence of language-bias in VidQA datasets, we retrieve a videos having a different answer for the same question. To facilitate research, we construct ActivityNet-SRL-QA and Charades-SRL-QA and benchmark them by extending three vision-language models. We perform extensive analysis and ablative studies to guide future work.

1 Introduction

Given a video, Video Question Answering (VidQA) requires a model to provide an answer to a video related question. Existing works treat VidQA as an N-way ($N \sim 1k$) classification task across a fixed set of phrases. Models trained under such formulations are strictly restricted in their recall rate, generalize poorly, and have severe limitations for end-user applications.

In this work, we introduce Video Question Answering with Phrases (VidQAP) which treats VidQA as a *fill-in-the-phrase* task. Instead of a question, the input to VidQAP consists of a query expression with a query-token. Then, given a video, VidQAP requires replacing query-token with a sequence of generated words. To generate a query, we leverage video descriptions and assign semantic roles to each phrase in these descriptions. Replacing a particular semantic-role with a query token produces a query-answer pair. We illustrate this in Figure 1 (details in Section 3.1).

While free-form answer generation is highly desirable, evaluating them is non-trivial due to two main challenges. First, existing language generation metrics like BLEU (Papineni et al. 2002) or BERTScore (Zhang et al. 2020) operate on sentences rather than phrases. When applied to short phrases, in the absence of context, even close matches like “A person” and “The man” would be falsely rejected due


			
Video description: A man on top of a building throws a bowling ball towards the pins			
Semantic Roles:	ARG0	V	ARG1 ARG2
<p>Q1: Who throws a bowling ball towards the pins? Model’s Top Predictions: A: A man B: A man under the tree C: A person D: This boy</p> <p>Correct Answer: A man on top of a building</p> <p>Q2: Does a man on top of a building throw a bowling ball towards to pins? Model’s Top Predictions: A: Yes B: No C: Maybe</p> <p>Correct Answer: Yes</p> <p>Q3: A man on top of a building throws a bowling ball towards the ____. Model’s Top Predictions: A: field B: pins C: basket D: man</p> <p>Correct Answer: pins</p> <p>(a) N-way Classification of Phrases</p>			
<p>Q4: <Q-ARG0> throws a bowling ball towards the pins. Model’s generated answer: A man standing on a house Correct answer: A man on top of a building</p> <p>Q5: A man on top of a building <Q-V> a bowling ball towards the pins. Model’s generated answer: throws Correct answer: throws</p> <p>Q6: A man on top of a building throws <Q-ARG1> towards the pins. Model’s generated answer: a ball Correct answer: a bowling ball</p> <p>Q7: A man on top of a building throws a bowling ball <Q-ARG2> Model’s generated answer: towards some bottles Correct answer: towards the pins</p> <p>(b) Free-form Answer Generation</p>			

Figure 1: Previous methods formulate VidQA as a N-way classification task. The questions are converted via question generation tool (Q1, Q2) or masking-out strategy (Q3). However, such QA has a theoretical recall upper bound when the correct answer is not among the choice list. In comparison, we propose a free-form text generation task which do not suffer such limitation (Q4-Q7)

to no n-gram overlap or poor contextual embeddings. Second, natural language questions often have strong language priors making it difficult to ascertain if the model retrieved information from the video.

To propose a reasonable evaluation metric, we revisit our *fill-in-the-phrase* formulation. Since we know where exactly the generated answer fits in the original query, we can create a complete sentence. With this key insight, we propose **relative scoring**: using the description as reference sentence, we

Dataset	Source	#Clips	Clip Duration(s)	#QA-Pairs	# QA / Clip	Task Type	Scripts	Box	QA Pair Creation
Movie-QA	Movies	6771	202.7	6462	0.95	MC	✓	✗	Human
Movie-FIB	Movies	128,085	4.8	348,998	2.72	OE	✗	✗	Automatic
VideoQA*	Internet videos	18100	45	174,775	9.66	OE	✗	✗	Automatic
MSVD-QA	Internet videos	1,970	9.7	50,505	25.64	OE	✗	✗	Automatic
MSR-VTT-QA	Internet videos	10,000	14.8	243,680	24.37	OE	✗	✗	Automatic
TGIF-QA	Tumblr GIFs	62,846	3.1	139,414	2.22	OE+MC	✗	✗	Human+Automatic
TVQA	TV Show	21,793	76	152,545	7	MC	✓	✗	Human
TVQA+	TV Show	4200	61.5	29,383	7	MC	✓	✓	Human
ActivityNet-QA*	Internet videos	5800	180	58000	10	OE	✗	✗	Human
ASRL-QA	Internet videos	35805	36.2	162091	5.54	OE + Phrase	✗	✓	Automatic
Charades-SRL-QA	Crowd-Sourced	9513	29.85	71735	7.54	OE + Phrase	✗	✗	Automatic

Table 1: Comparison of Existing datasets for VidQA with our proposed ASRL-QA and Charades-SRL-QA. Here, OE = Open-Ended, MC = Multiple Choice. “Scripts”: if answering questions requires access to scripts or subtitles. “Box”: if dataset provides bounding box annotations. *: Includes Yes/No questions

compute the metrics once by replacing the query-token once with the predicted answer phrase and once with an empty-string. The model’s performance is measured by the relative improvement from the predicted answer compared to the empty string. In particular, substituting the answer phrase in the query expression allows the computing the contextual embeddings required by BERTScore.

To mitigate the language-bias issue, we emulate the procedure proposed by (Goyal et al. 2017) where for a given question, another image (or video in our case) is retrieved which has a different answer for the same question. To retrieve such a video, we use a contrastive sampling method (Sadhu, Chen, and Nevatia 2020) over the dataset by comparing only the lemmatized nouns and verbs within the SRLs. We then propose **contrastive scoring** to combine the scores of the two answer phrases obtained from the contrastive samples (details on evaluation in Section 3.2).

To investigate VidQAP, we extend three vision-language models namely, Bottom-Up-Top-Down (Anderson et al. 2018), VOGNet (Sadhu, Chen, and Nevatia 2020) and a Multi-Modal Transformer by replacing their classification heads with a Transformer (Vaswani et al. 2017) based language decoder. To facilitate research on VidQAP we construct two datasets ActivityNet-SRL-QA (ASRL-QA) and Charades-SRL-QA and provide a thorough analysis of extended models to serve as a benchmark for future research (details on model framework in Section 3.3 and dataset creation in Section 4.1).

Our experiments reveal that there exists a large disparity in performance across semantic-roles (i.e. queries for some roles can be answered very easily compared to other roles). Moreover, certain roles hardly benefit from vision-language models suggesting room for improvement. Finally, we investigate the effects of relative scoring and contrastive scoring for VidQAP with respect to BertScore.

Our contributions in this work are two-fold: (i) we introduce VidQAP and propose a systematic evaluation protocol to leverage state-of-art language generation metrics and reduce language bias (ii) we provide extensive analysis and contribute a benchmark on two datasets evaluated using three vision-language models. We will release the dataset and code upon publication.

2 Related Works

Question Answering in Images has received extensive attention in part due to its end-user applicability. Key to its success has been the availability of large-scale curated datasets like VQA v2.0 (Goyal et al. 2017) for visual question answering and GQA (Hudson and Manning 2019) for relational reasoning. To address the strong language priors, the datasets are balanced by retrieving images which given the same question lead to a different answer. However, these procedures cannot be extended for VidQA since crowd-sourcing to retrieve videos is expensive and there exists no scene-graph annotations for videos. In this work, we perform the retrieval using lemmatized nouns and verbs of the semantic roles labels obtained from video descriptions to balance the dataset.

Question Answering in Videos: has garnered less attention compared to ImageQA. A major bottleneck is that there is no principled approach to curating a VidQA dataset which reflects the diversity observed in ImageQA datasets. For instance, naively crowd-sourcing video datasets leads to questions about color, number which is same as ImageQA datasets and doesn’t reflect any spatial-temporal structure. To address this issue, TGIF-QA (Jang et al. 2017) and ActivityNet-QA (Yu et al. 2019) use a question-template to enforce questions requiring spatio-temporal reasoning but forgo the question diversity. An orthogonal approach is to combine VidQA with movie scripts (Tapaswi et al. 2016) or subtitles (Lei et al. 2018). However, this severely restricts the domain of videos. Moreover, recent works have noted that language-only baselines often outperform vision-language baselines (Jasani, Girdhar, and Ramanan 2019; ning Yang et al. 2020; Zellers et al. 2019).

Automatic Question Generation: Due to the above limitations, the dominant approach to create large-scale VidQA dataset has been automatic question generation from existing video descriptions which can be easily crowd-sourced. Our proposed formulation of using SRLs to generate query-expressions falls in this category. Prior works include VideoQA (Zeng et al. 2017), MSR-VTT-QA and MSVD-QA (Xu et al. 2017) which use a rule based question generator (Heilman and Smith 2009) to convert descriptions to questions and Movie-Fill-in-the-Blanks (Maharaj et al.

ARG0 V ARG1 DIR LOC
A person moves exercise equipment around in the office

Query-Expressions	Answers
<Q-ARG0> moves exercise equipment in the office	A person
A person <Q-V> exercise equipment in the office	moves
A person moves <Q-ARG1> in the office	exercise equipment
A person moves exercise equipment <Q-LOC>	in the office

(a) Following SRLs are considered: **ARG0**, **ARG1**, **ARG2**, **V**, **LOC** to generate query-expressions and answers. Here, the phrase corresponding to the semantic-role **DIR** is removed from both query-expressions and answers.

ARG0 V DIR MNR
A person climbs down with his hands folded

(b) Query-expressions would have less than 3 semantic-roles and hence ignored.

Figure 2: Illustration of our query generation process. In (a) **DIR** is ignored from both Query and Answers. In (b) the question is removed from validation set since at most two arguments from considered set are present.

2017) which mask out at most one word which could be a noun, adjective or verb in a sentence. In comparison, our method poses VidQAP as fill-in-blanks but with phrases, explicitly asks questions about actions, and the answer phrases are not constrained to a fixed set. As a result of this increased space of phrases, methods on existing datasets cannot be directly applied to VidQAP. To enable further research, we contribute two datasets ASRL-QA and Charades-SRL-QA. In Table 1 we compare these with existing VidQA datasets.

SRL in Vision: has been explored in the context of human object interaction (Gupta and Malik 2015), situation recognition (Yatskar, Zettlemoyer, and Farhadi 2016), and multimedia extraction (Li et al. 2020). Most related to ours is the usage of SRLs for grounding (Silberer and Pinkal 2018) in images and videos (Sadhu, Chen, and Nevatia 2020). Our work builds on (Sadhu, Chen, and Nevatia 2020) in using SRLs on video descriptions, however, our focus is not on grounding. Instead, we use SRLs primarily as a query generation tool and use the argument as a question directive.

3 Design Considerations for VidQAP

The VidQAP task is conceptually simple: given a video and a query expression with a query-token, a model should output an answer phrase that best replaces the query-token. This leads to three main design considerations: (i) How to generate a query-expression from existing resources (Section 3.1) (ii) How to evaluate the answer phrases returned by a model (Section 3.2) (iii) What modeling framework choices enable VidQAP (Section 3.3).

3.1 Using SRLs to Generate Queries for VidQAP

We first briefly describe semantic-role labels (SRLs)¹. Then we detail how SRLs are used to create VidQAP queries.

¹Detailed discussion is provided in supplementary. A demo is available here: <https://demo.allennlp.org/semantic-role-labeling>

Query Generation Using SRLs: Semantic Role Labels (SRLs) provide a high-level label to entities extracted from a sentence in the form of who (ARG0), did what (V) to whom (ARG1) (Strubell et al. 2018). Other roles such as to whom / using what (ARG2) and where (LOC) are also common. As a pre-processing step, we assign SRLs to video descriptions using a state-of-art SRL labeler (Shi and Lin 2019). A particular description could consist of multiple verbs, in which case, we consider each verb and its associated SRLs independently. For a particular semantic-role, we substitute the corresponding phrase with a query token to generate the query expression. The replaced phrase is the corresponding answer. Using this method we are able to generate multiple queries from a single description. A complementary advantage of using SRLs is that query phrases are centered around “verb-phrases” which are relevant to the video contents.

Generating queries using every SRL is not beneficial as some SRLs have more to do with the phrasing of the language rather than the video. For instance, in the phrase “Players are running around on the field”, if we mask out the word “around” (DIR), it can be answered without looking at the video. To address the above issue, we confine our description phrases to a fixed set of semantic-roles namely: ARG0, ARG1, V, ARG2, ARGM-LOC. Only those phrases which belong to the above set of SRLs may appear in the query-expression or as an answer phrase. We further remove phrases which have only two arguments as these are too ambiguous to fill. Figure 2 illustrates these steps.

3.2 Evaluating Answer Phrases

A key challenge in VidQAP is the lack of any standard protocol to evaluate free-form generated phrases. A simple way is to adopt metrics like BLEU (Papineni et al. 2002), ROUGE (Lin 2004), METEOR (Banerjee and Lavie 2005), and CIDER (Vedantam, Zitnick, and Parikh 2015) which are already used for captioning in images and videos. However, these metrics suffer from limited generalization: BLEU, ROUGE, and CIDER require exact n-gram matches. While this is fine for captioning where longer phrases average out errors, answers phrases are typically much smaller than a complete sentence. This can lead to many correct answers receiving very low score.

This issue is resolved to a certain extent for captioning by learned metrics like BERTScore (Zhang et al. 2020) which utilize contextual embeddings obtained from large pretrained models like BERT (Devlin et al. 2019) and RoBERTa (Liu et al. 2019). However, answer phrases are usually short and don’t provide meaningful contextual embeddings. In the extreme case when the answer is a single word, for instance when the query is about a **Verb**, these embeddings turn out to be very noisy leading to large number of false-positives.

Relative Scoring: To enable usage of contextual embeddings, we propose evaluating the relative improvement of the generated answer phrase compared to the ground-truth phrase. We denote the input query expression as Q , the ground-truth answer is A_{gt} , and the predicted answer is A_{pred} . Let $Q(X)$ denote Q with the question tokens replaced by X . Then for a given metric B , we compute the

Query Expression:	A person <Q-V> exercise equipment.
Reference (Ground Truth):	A person moves exercise equipment.
Hypothesis (Prediction):	A person lifts exercise equipment.
Baseline (Empty String):	A person exercise equipment.

$$\alpha = B(\text{Ref}, \text{Base}), \quad \beta = B(\text{Ref}, \text{Hyp}), \quad \gamma = B(\text{Ref}, \text{Ref})$$

$$\text{Relative Metric Score } B_r(\text{Ref}, \text{Hyp}) = \frac{\beta - \alpha}{\gamma - \alpha}$$

Figure 3: Illustration of the Relative Metric Computation. “moves” is the ground-truth answer and “lifts” is a model’s prediction. The Relative Metric compares the relative improvement from using the model’s prediction compared to an empty string.

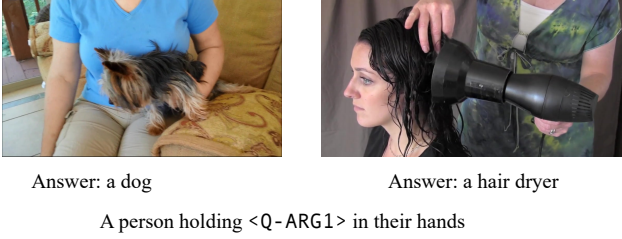


Figure 4: Illustration of Contrastive Sampling Process. For the same query-expression, we retrieve two videos with different answers. The model should be able to answer both queries.

relative metric B_r as (see Figure 3 for illustration)

$$\text{Ref} = Q(A_{gt}) \quad \text{Hyp} = Q(A_{pred}) \quad \text{Base} = Q("") \quad (1)$$

$$B_r(A_{gt}, A_{pred}) = \frac{B(\text{Ref}, \text{Hyp}) - B(\text{Ref}, \text{Base})}{B(\text{Ref}, \text{Ref}) - B(\text{Ref}, \text{Base})} \quad (2)$$

Note that $B(\text{Ref}, \text{Ref})=1$ for BLEU, METEOR, ROUGE, BERTScore but not for CIDEr.

We observe that Eqn 2 is very similar to the re-scaling proposed in BERTScore. However, in BertScore re-scaling aims at making the score more readable and doesn’t change the relative ranking of the hypothesis. In our case, Eqn 2 plays two roles: first, it allows computing the contextual embeddings because the answers are now embedded inside a complete phrase, second while the ranking is not affected for a particular query, the score would be different across queries and hence affect the overall relative metric.

Contrastive Scoring: Visual Question Answering suffers from heavy language priors, and as a result, it is often difficult to attribute whether the image or video played a role in the success. For images, (Goyal et al. 2017) resolved this by balancing the dataset where they crowd-sourced the task of collecting an image that has a different answer for the same question. However, such a crowd-sourcing method is difficult to extend to videos since searching for videos requires a much longer time. This is further complicated by accepting answer phrases compared to single word.

We simulate the balancing process using the contrastive sampling method used in (Sadhu, Chen, and Nevatia 2020). Specifically, for a given video-query-answer (V_1, Q_1, A_1) tuple we retrieve another video-query-answer (V_2, Q_2, A_2)

tuple which share the same semantic role structure as well as lemmatized noun and verbs for the question, but a different lemmatized noun for the answer. At test time, the model evaluates the question separately, but the evaluation function requires both answers to be correct. Since our answers comprise of phrases, the notion of correctness is not absolute (unlike say accuracy metric). Thus, we put a threshold t below which the answer is deemed incorrect.

Mathematically, let $S_i = B_r(A_{gt_i}, A_{pred_i})$ be the relative score for sample i , and we are given sample j is a contrastive example for sample i . Then the contrastive score (CS_i) for sample i at a threshold T_{CS} would be

$$CS_i = \max(S_i \mathbb{1}[S_j > T_{CS} * B(\text{Ref}_j, \text{Ref}_j)], 0) \quad (3)$$

Here $\mathbb{1}[]$ is the indicator variable which is 1 if the expression within brackets is True, otherwise 0. The \max operator ensures the scores don’t become negative. For our experiments, we use $T_{CS}=0$ which requires that the answer for the contrastive sample should be better than an empty string.

We further use the contrastive samples to compute a consistency metric. For sample i , the consistency $Cons_i$ for a threshold T_{cons} is given by

$$Cons_i = \mathbb{1}[(S_i - T_{cons}) * (S_j - T_{cons}) > 0] \quad (4)$$

In other words, Consistency requires the model to be either correct or incorrect for both contrastive samples.

Combined Metric at a Glance: Given a metric B , for a given sample i and its contrastive sample j

1. Compute the relative metric (Eqn 2) for both i, j
2. Compute contrastive Score for the metric (Eqn 3)
3. Optionally compute Consistency for the metric (Eqn 4)

We use the prefix “R-” such as R-B to denote both relative scoring and contrastive scoring is being computed. We report Consistency for BertScore with $T_{cons}=0.1$

3.3 Model Framework

Models for VidQAP require a language encoder to encode the question, a visual encoder to extract video features and a decoder to generate a sequence of words.

Inputs include query expression $\{w\}_{i=1}^L$ (L is number of words), video segment features for F_1 frames and optionally k RCNN features for F_2 frames. In either case, frames are sampled uniformly from the video segment time-span. While the models differ in their encoding scheme, our language decoder model (Transformer based) used to generate the output answer phrase is kept same across all models.

Lang-QAP: is a language-only model and only uses the query input. It uses Transformer based encoder to encode the query into $\hat{q} \in \mathbb{R}^{L \times d}$. The decoder only uses last layer output of the encoder (Figure5-(a)).

BUTD-QAP: Bottom-up-Top-Down (Anderson et al. 2018) is a popular approach for image question answering as well as captioning. It first computes attention between the question and the RCNN visual features to generate an attended visual feature, which is then used with the question to produce an output answer. Here, we replace the RCNN features with the segment features ($\hat{v} \in \mathbb{R}^{F_1 \times d}$). We can also

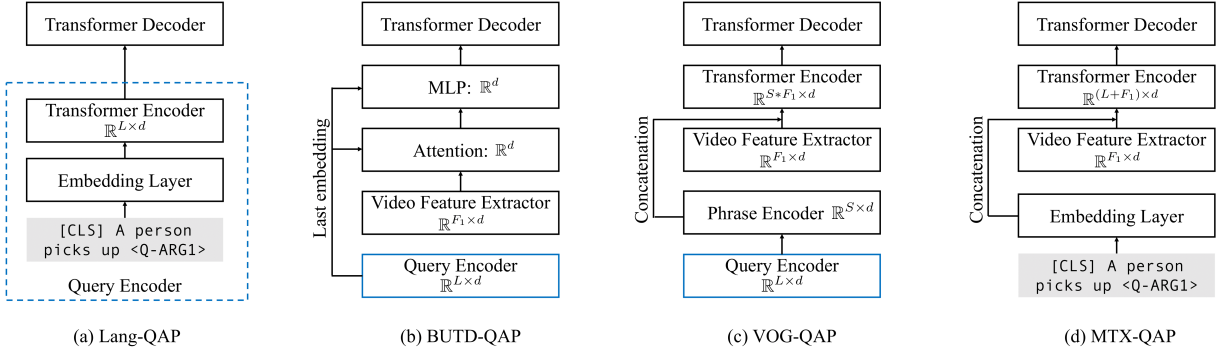


Figure 5: Schematic of the various models used to benchmark VidQA. Input Query: “A person picks up <Q-ARG1>”. Ground-Truth Answer: “a pair of shoes”. (a) Lang-QAP is a language-only model which encodes the query input and passes to a decoder. (b) BUTD-QAP uses the pooled feature representation from language encoder and attends over the visual features. (c) VOG-QAP uses an additional phrase encoder and applies a Transformer over the multi-modal features (d) MTX-QAP consumes both the language and visual features with a multi-modal transformer.

include RCNN features by projecting them to same dimension as segment features and then concatenate them along the frame-axis ($\hat{v} \in \mathbb{R}^{(F_1+F_2 \times k) \times d}$). For language features, we use the [CLS] token representation obtained from the last layer of the language encoder used in Lang-QAP. The final output using the language and visual features is ($\hat{m} \in \mathbb{R}^d$) passed to the decoder (Figure 5 (b)).

VOG-QAP: VOGNet (Sadhu, Chen, and Nevatia 2020) has been proposed for grounding objects in videos given a natural language query. We first derive per phrase encoding which corresponds to a single SRL i.e. $\hat{q} \in \mathbb{R}^{S \times d}$ (S is number of semantic roles) and concatenate them with the visual features which are same as those used in BUTD-QAP (i.e. \hat{v}), to get multi-modal features $m[l, i] = [\hat{v}_i || \hat{q}_l]$ and reshape it to get $m \in \mathbb{R}^{S \times F \times d}$. These multi-modal features are used to generate the output sequence (Figure 5 (c)).

MTX-QAP: Recently, transformer models pre-trained on large-scale paired image-text data have become popular. Even in the absence of pre-training, such architectures can achieve competitive performance (Lu et al. 2019). In the context of videos, ActBert (Zhu and Yang 2020) has been proposed. We create a similar architecture to ActBert but we replace their proposed Tangled-Transformer with a vanilla Transformer². Specifically, we jointly encode the language and visual features in a single transformer and feed the output to the decoder (Figure 5 (d)).

Training: All models are trained using smoothed label cross-entropy with teacher forcing.

4 Experiments

We briefly discuss the dataset creation process (Section 4.1), followed by experimental setup (Section 4.2). We then summarize our results (Section 4.3) and discuss key-findings. We provide qualitative visualizations of our dataset, metrics and trained models in the supplementary material.

²The code for ActBert is not publicly available.

4.1 Dataset Creation and Statistics

We create two datasets ASRL-QA and Charades-SRL-QA derived from ActivityNet-SRL (Sadhu, Chen, and Nevatia 2020) and Charades (Sigurdsson et al. 2016) respectively.

There are three key steps to create QA datasets from descriptions: (i) assign semantic-roles to the descriptions (ii) perform co-reference resolution so that the questions are self-contained (iii) obtain lemmatized nouns and verbs to perform contrastive sampling. We follow (Sadhu, Chen, and Nevatia 2020) and use (Shi and Lin 2019) for semantic-role labeling. For co-reference resolution, we use the co-reference resolution model provided by allennlp library (Gardner et al. 2017) which uses the model by (Lee et al. 2017) but replaces the GloVe (Pennington, Socher, and Manning 2014) embeddings with SpanBERT embeddings (Joshi et al. 2019)³.

Since Charades primarily involves videos with a single person, we discard questions involving ARG0. We limit to using a single description per video to avoid repetitive questions. We re-use the same train split for both datasets. For ASRL-QA, since test set of ActivityNet is not public and Charades only has a test set but no official validation set. Thus, we split the existing validation set by video names and create the validation and test sets. For both validation and test splits, we remove those questions for which no contrastive sample was found as it indicates data-biases.

4.2 Experimental Setup

Dataset Statistics: ASRL-QA consists of 35.7k videos and 162k queries split as training, validation and testing as 30.3k, 2.7k, 2.7k videos and 147k, 7.5k, 7.5k queries. Note that the size of validation and test set are proportionately smaller as we include only those queries which have a corresponding contrastive sample, whereas no such filtering is done for the train set (nearly 95k queries in train set have a contrastive

³<https://demo.allennlp.org/coreference-resolution>

	ASRL-QA						Charades-SRL-QA					
	R-BS	Cons	R-B@2	R-R	R-M	R-C	R-BS	Cons	R-B@2	R-R	R-M	R-C
Lang-QAP	0.402	0.728	0.228	0.182	0.125	0.095	0.406	0.719	0.277	0.253	0.147	0.121
BUTD-QAP	0.413	0.716	0.237	0.203	0.147	0.105	0.399	0.714	0.271	0.231	0.115	0.105
VOG-QAP	0.414	0.717	0.239	0.204	0.142	0.108	0.442	0.739	0.297	0.274	0.165	0.136
MTX-QAP	0.414	0.715	0.247	0.206	0.149	0.113	0.439	0.757	0.294	0.267	0.157	0.139

Table 2: Comparison of our extended models for VidQAP across two datasets on our proposed Metric. Here, “R-” prefix implies it is the final metric computed after relative scoring and contrastive scoring with threshold 0. “BS”: BertScore, “Cons”: Consistency on BertScore, B@2: Sentence BLEU-2, R: ROUGE, M: METEOR, C: CIDEr. All reported numbers are on the test set.

	ASRL-QA					Charades-SRL-QA			
	ARG0	V	ARG1	ARG2	LOC	V	ARG1	ARG2	LOC
Lang-QAP	0.697	0.519	0.325	0.322	0.145	0.631	0.458	0.33	0.206
BUTD-QAP	0.681	0.515	0.372	0.334	0.162	0.568	0.413	0.316	0.299
VOG-QAP	0.671	0.513	0.366	0.332	0.188	0.63	0.467	0.365	0.305
MTX-QAP	0.702	0.478	0.374	0.344	0.17	0.633	0.455	0.364	0.304

Table 3: Comparison of our extended models per SRL. All reported scores are R-BS: BertScore computed after relative scoring and contrastive scoring with threshold 0.

	R-BS	Cons
LangC	0.253	0.889
LangC (no SW)	0.103	0.943
MTxC	0.254	0.869
MTxC (no SW)	0.103	0.939

Table 4: Comparison of models using N-way classification across top-1k phrases. no SW: stop words are removed on ASRL-QA

pair). Charades-SRL-QA consists of 9.4k videos and 71.7k queries split as training, validation and testing as 7.7k, 0.8k, 0.8k videos and 59.3k, 6.1k, 6.2k queries. Despite its smaller size, the validation and test set of Charades-SRL-QA is similar to ASRL-QA as Charades is curated with the goal of diversifying subject, verb, object tuples. More details about the data statistics and visualizations are in supplementary.

Evaluation Metrics: As discussed in Section 3.2, we report the combined metric (i.e. metrics prefixed with “R-”) for the commonly used generation metrics: BLEU, METEOR, ROUGE, CIDEr and BertScore. We use the evaluation metrics used in COCO-Captions (Chen et al. 2015). For BLEU, we report the sentence level BLEU-2.

Implementation Details: For ASRL-QA we use the same segment and proposal features provided by (Zhou et al. 2019) for ActivityNet-Entities. Segment features are computed at every 0.5 second of a video using Temporal Segment Networks (Wang et al. 2016). Proposal features are computed at 10 frames uniformly sampled from the video segment using FasterRCNN (Ren et al. 2015) trained on visual genome (Krishna et al. 2016). We use 5 proposals per frame which is the same as *GT*5 setting proposed in (Sadhu, Chen, and Nevatia 2020). For Charades-SRL-QA segment features we use the S3D (Xie et al. 2018) model pre-trained on the HowTo100M dataset (Miech et al. 2019, 2020). We don’t use proposal features for Charades.

Our models are implemented in Pytorch (Paszke et al. 2019). For Transformers, we use the implementation provided by Fairseq (Ott et al. 2019). For both transformer encoders and decoders, we use 3 layers with 8 attention heads. The decoder is trained using teacher-forcing on the answer phrases. We use $d=512$ as the hidden embedding size for both visual features and language features.

We train our models for 10 epochs with batch size of 32

with learning rate $1e^{-4}$ and use the model with highest validation metric (we use R-BertScore) for testing. Complete implementation details are provided in the supplementary.

4.3 Results and Discussions

In Table 2 we compare the performance of the proposed VidQAP models with a language-only baseline on two datasets ASRL-QA and Charades-SRL-QA. As mentioned in Section 3.2 the prefix “R-” denotes the combined metric of relative scoring followed by contrastive scoring.

Comparing Metrics: It is evident that compared to other metrics, R-BertScore shows a higher relative improvement. This is because BertScore allows soft-matches by utilizing contextual embeddings obtained from a pre-trained BERT (Devlin et al. 2019) or Roberta (Liu et al. 2019) model.

Comparison Across Datasets: We find that performance on both the datasets follow very similar trends for all metrics. Charades-SRL-QA has slightly higher scores compared to ASRL-QA likely because it has lesser data variations (Charades is mostly confined indoor videos). This is encouraging as it suggests findings on one either dataset would transfer.

Lower Performance of BUTD on Charades-SRL-QA: We observe that BUTD-QAP performs worse than language-only baseline on Charades-SRL-QA but outperforms on ASRL-QA. We hypothesize that Charades has subtle clues in the query-expression which are picked up by the language encoder when the whole sequence is considered but is missed when only the pooled hidden representation of the query is considered.

Comparison Across Models: Other than the previous exception, we find that multi-modal models outperform language-only baseline. However, the improvement over language baseline is small. To understand why the perfor-

mance gap is small, in Table 3 we report R-BertScore for every considered SRL.

We find a large disparity in performance depending on the SRL. Most strikingly, multi-modal models perform worse than language-only model on ARG0 and V. For ARG0, the strong performance of the Lang-QAP arises because most of the time the agent who causes an action is a human. Therefore answer phrases having simply “A man” or “A woman” or “A person” leads to reasonable performance. This additionally suggests that grounding “who” is performing the action remains a non-trivial task.

The more surprising result is the strong performance of Lang-QAP on V which is consistent across both datasets despite using contrastive sampling. There are two likely causes. First, the distinction between verbs is not as strict as object nouns and as a result many similar verbs are classified as a separate verb and thereby diminishing the returns of contrastive sampling. For instance, “jumping” and “hoping” have different lemma and thus considered distinct verbs but R-BS would treat them as similar even if the specific action would be classified “jumping” rather than “hoping”. Second, the existence of other SRLs such as ARG1 confines the set of possible verbs. For instance, if the object is “glass”, only limited verbs such as “drink”, “hold” are probable.

On the remaining arguments namely ARG1, ARG2, and LOC, multi-modal models show a significant improvement over language-only baseline ranging from 1–10%. However, the performance on absolute terms remains very low. As such, our proposed task VidQAP remains extremely challenging for current multi-modal models with a healthy gap remaining to be filled.

Comparison with N-way Classification: We investigate the advantages of using a decoder network to generate phrases compared to an N-way classification over a fixed set of phrases. For fair comparisons, we keep the entire encoding network the same. We simply replace the decoding network with a classifier, which we denote with the suffix “C”. To create the fixed set of phrases, we experiment with two settings: one including stop-words and other excluding stop-words. In both cases, we obtain the top-1000 phrases from the training set, and train the in the exact same manner as their decoder counterpart. It is evident from Table 4 that N-way classification achieves a sub-par result by a significant margin of over 14% point. These achieve higher consistency due to limited number of choices.

Evaluation Metric Scores: In Table 5 we record the BertScore computation in three parts: directly computing over the answer phrases, performing relative scoring, finally performing contrastive scoring with different thresholds.

We observe that for V, naive computation leads to absurdly high scores. This is because verbs consist of a single word and thus the embeddings are not contextual. This is remedied by relative scoring and is further controlled by combining with contrastive sampling.

Further note that relative scoring operates differently based on the SRLs. For instance, it increases the score for ARG0 and ARG1 where the answers more often paraphrased the ground-truth questions while for ARG2 and LOC, it decreases the score due to incorrect matches. While contrastive

		ARG0	V	ARG1	ARG2	LOC
Lang-QAP	Direct	0.552	0.9268	0.234	0.302	0.216
	Rel Score	0.7	0.534	0.332	0.237	0.1
	CS@0	0.697	0.519	0.325	0.322	0.145
	CS@0.1	0.69	0.492	0.295	0.28	0.132
	CS@0.2	0.68	0.459	0.262	0.212	0.106
	CS@0.3	0.657	0.423	0.219	0.149	0.085
MTX-QAP	Direct	0.566	0.929	0.269	0.321	0.258
	Rel Score	0.706	0.488	0.366	0.25	0.14
	CS@0	0.702	0.478	0.374	0.344	0.17
	CS@0.1	0.693	0.45	0.343	0.305	0.145
	CS@0.2	0.681	0.413	0.306	0.239	0.117
	CS@0.3	0.659	0.376	0.27	0.17	0.08

Table 5: BertScore Metrics computed Directly on answer phrases. Rel Score: After Relative Scoring. CS@T: Contrastive scoring with threshold T.

	ARG0	V	ARG1	ARG2	LOC	Overall
BUTD-QAP	0.706	0.506	0.388	0.36	0.196	0.431
VOG-QAP	0.704	0.516	0.366	0.352	0.202	0.429
MTX-QAP	0.685	0.465	0.378	0.355	0.19	0.416

Table 6: Effect of Adding Region Proposals. All reported scores are R-BS

scoring is aimed at reducing language-only bias and as such should always reduce the relative score, we observe increased score in ARG2 for both Lang-QAP and MTX-QAP. This is caused by the *max* function which restricts the lower-limit to be 0.

Effect of Region Boxes: As noted earlier, the visual features can also include region features extracted from an object detector like FasterRCNN (Ren et al. 2015). In Table 6 we record the effect of including regional features. In particular, we use the GT5 setting used in (Sadhu, Chen, and Nevatia 2020) where 5 region proposals are used from 10 frames uniformly sampled from the video segment. Interestingly, MTX-QAP under-performs than both BUTD-QAP and VOG-QAP on ARG0. A possible reason is that the transformer is unable to effectively reason over both language and vision over such a large range of inputs.

5 Conclusion

In this work, we introduce Video Question Answering with Phrases (VidQAP) where we pose VidQA as a fill-in-the-phrases task. Given a video and query expression, a model needs to compose a sequence of words to answer. We then propose a method to leverage semantic roles from video descriptions to generate query expressions and outline a robust evaluation protocol. This involves computing the relative improvement of the prediction answer compared to an empty string followed by a contrastive sampling stage which reduces language-only biases. We then contribute two datasets ASRL-QA and Charades-SRL-QA to facilitate further on VidQAP and benchmark them with three vision-language models extended for our proposed task.

Appendix

This is the appendix for the paper “Video Question Answering with Phrases via Semantic Roles”. The appendix provides details on

1. Dataset construction and Dataset statistics (Section A)
2. Implementation Details for both the Metrics as well as the Models (Section B).
3. Visualization of Model Outputs (Section C)

A Dataset Construction

We first discuss semantic-role labeling used in natural language processing. Then, we detail the dataset construction process used for ASRL-QA and Charades-SRL-QA (Section A.2) and then provide the dataset statistics (Section A.3).

A.1 Semantic Role Labeling

Semantic-Role Labels extract out high-level meanings from a natural language description. Two widely used SRL annotations are PropBank (Kingsbury and Palmer 2002) and FrameNet (Baker, Fillmore, and Lowe 1998). Here we use SRLs which follow PropBank annotation guidelines (see (Bontal et al. 2012) for complete guideline).

Most commonly used argument roles are

- V: the verb. All remaining roles are dependent on this verb. While the numbered arguments differ slightly based on the verb used, they share common themes across verbs as listed below (see (Bontal et al. 2012) for full details). For instance, “cut” is a Verb.
- ARG0: the agent, or the one causing the verb. For most action verbs, this is usually a human or an animal. For instance, “A person cuts a vegetable”, “A person” is ARG0.
- ARG1: the object, on which the action is being performed. In “A person cuts a vegetable”, “a vegetable” is ARG1.
- ARG2: the tool being used for the verb, or someone who benefits from the verb. For instance, in “A person is cutting a vegetable with a knife”, “with a knife” denotes the tool and is ARG2. In “A person throws a basketball to the basket”, “to the basket” denotes the benefactor and is ARG2.
- ARG-M-LOC or simply LOC denotes the place or location where the verb takes place. For instance, in “A person is cutting a vegetable on a plate”, “on a plate” is the LOC.

These SRLs form the basis of our dataset construction process. To assign SRLs to language descriptions we use allennlp library (Gardner et al. 2017) which provides an implementation of a BERT (Devlin et al. 2019) based semantic-role labeler (Shi and Lin 2019). The system achieves 86.49 F1 score on OntoNotes (Pradhan et al. 2013) 5.0 dataset.

A.2 Construction Process

Both ASRL-QA and Charades-SRL-QA follow the same process with few subtle differences. For both datasets:

1. Pre-Process Data:
 - Assign semantic role labels (SRLs) to video descriptions using SRL labeller (Shi and Lin 2019).

- Remove stopword verbs with lemmas: “be”, “start”, “end”, “begin”, “stop”, “lead”, “demonstrate”, “do”.
- For the original descriptions spread across multiple video segments, combine the sentences into a document. Use a co-reference resolution model on this model (we use (Lee et al. 2017) with SpanBERT embeddings (Joshi et al. 2019) provided in allennlp library (Gardner et al. 2017)).
- Replace the following pronouns: “they”, “he”, “she”, “his”, “her”, “it” with the relevant noun-phrase obtained from the co-reference resolution output.

2. Query-Generation:

- For each verb-role set within a description (each description can have multiple verbs), consider the role set ARG0, ARG1, V, ARG2, LOC for ASRL-QA and ARG1, V, ARG2, LOC for Charades-SRL-QA.
- If there are at least 3 verb-roles for the given verb, for each SRL replace it with a query token (with $\langle Q-\{R\} \rangle$ where R is the role). This forms one query. Repeat for all SRLs in the considered set.
- The minimum of 3 verb-roles is present to avoid ambiguity in the query. Limiting the argument role-set helps in generating queries less likely to have strong language-priors (though as seen in qualitative examples, some priors are still present).
- After the queries are generated, create lemmatized verbs, and nouns set for each query, and store the video segment ids in a dictionary. This is similar to the process used in (Sadhu, Chen, and Nevatia 2020), with the difference that we additionally have query-tokens.
- For each query, use the dictionary to sample set of video segment ids which share the same semantic role structure, but for the query-token have a different answer. These are used for matching when computing the scores for the validation and testing set using the contrastive score.

3. Creating Train/Test Splits:

- Keep the training set for each dataset the same.
- For validation and testing, we split the dataset based on the video ids (half video ids are set as validation, and half as testing). The queries are then split based on the video ids.
- Note that while contrastive sampling is done before validation test split. So validation and test ids are used for computing the other’s score for contrastive sampling. This is similar to the setting used in (Sadhu, Chen, and Nevatia 2020) as the total number of videos available for validation, and testing are insufficient for contrastive sampling.

A.3 Dataset Statistics

Dataset statistics can be found in Table 1. Lemma distributions are visualized in Figure 1 Overall, we find slightly skewed distribution of Argument roles across the datasets. For instance, ARG0, ARG1 are much more frequent than ARG2 and LOC. Also, since every SRL needs to have a verb (V), the distribution of the videos is the same as the overall.

		ASRL-QA			Charades-SRL-QA		
		Train	Val	Test	Train	Val	Test
Overall	Videos	30337	2729	2739	7733	860	876
	Queries	147439	7414	7238	59329	4431	4520
	Query Length	8.03	6.03	6	7.11	5.6	5.62
	Answer Length	2.2	2.33	2.33	1.83	1.96	1.94
ARG0	Videos	24483	1372	1419	NA		
	Queries	37218	1603	1643			
	Query Length	7.31	5.73	5.65			
	Answer Length	2.51	2.37	2.48			
V	Videos	29922	1737	1733	7733	802	811
	Queries	52447	2247	2187	27745	1824	1829
	Query Length	9.2	7.26	7.18	7.7	6.37	6.44
	Answer Length	1	1	1	1	1	1
ARG1	Videos	24863	1810	1793	7600	808	828
	Queries	36787	2250	2179	21557	1857	1874
	Query Length	7.4	5.4	5.43	6.43	5.07	5.04
	Answer Length	2.8	2.82	2.83	2.31	2.39	2.39
ARG2	Videos	12048	850	805	5433	490	522
	Queries	14321	941	886	8279	651	699
	Query Length	7.49	5.45	5.36	6.94	5.13	5.13
	Answer Length	3.55	3.69	3.62	3.11	3.22	3.04
LOC	Videos	6025	340	319	1578	87	112
	Queries	6666	373	343	1748	99	118
	Query Length	7.57	5.17	5.35	6.93	4.75	5.06
	Answer Length	3.61	3.87	3.63	3.22	3.19	3.08

Table 1: Detailed dataset statistics for both ASRL-QA and Charades-SRL-QA with respect to different argument roles. Recall that ARG0 is not present in Charades-SRL-QA, and hence the corresponding rows are kept blank.

B Implementation Details

We first report the implementation details for the metrics (Section B.1). Then, we detail the model implementation details (Section B.2).

B.1 Metric Implementation

For Bleu (Papineni et al. 2002), Rouge (Lin 2004), Meteor (Banerjee and Lavie 2005), and CIDEr (Vedantam, Zitnick, and Parikh 2015) we use the implementations provided in coco-captions repository⁴ (Chen et al. 2015).

For BERTScore we use the official implementation⁵

BLEU-2: computes Bleu with n-gram with $n=2$. We use sentence-bleu score instead of the more commonly used corpus bleu score. This is further used for contrastive sampling.

ROUGE: we use ROUGE-L which computes the longest common sub-sequence.

METEOR: we use Meteor 1.5 version (Denkowski and Lavie 2014).

CIDEr: we use CIDEr-D implementation which includes idf-weighting.

BertScore: we use BertScore with hash “roberta-large_L17_idf_version=0.3.5(hug_trans=3.0.2)-rescaled”

We show examples of computing the metrics.

B.2 Model Implementation

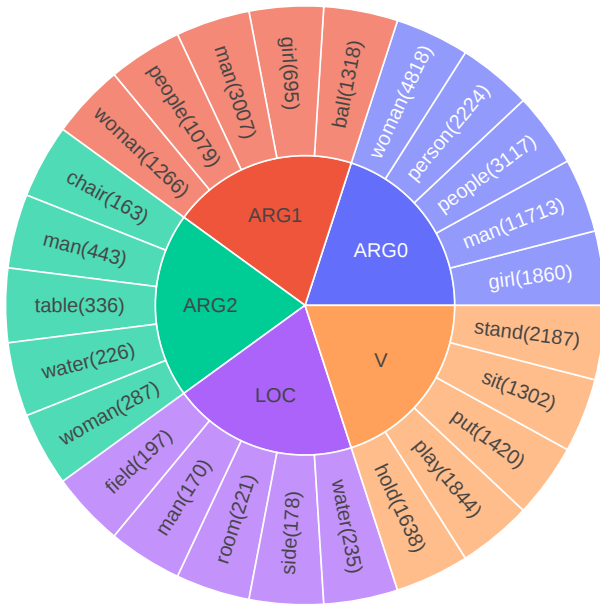
We report all model implementation details.

General Settings: Our code is implemented using Pytorch (Paszke et al. 2019). For Transformer, we use the implementation provided in FairSeq (Ott et al. 2019). The vocabulary consists of 5k words for ASRL-QA and 3k words for Charades-SRL-QA. The segment features are of dimension 3072 and 512 for ASRL-QA and Charades-SRL-QA respectively obtained from TSN (Wang et al. 2016) and S3D (Xie et al. 2018) trained on HowTo100M (Miech et al. 2019) using the loss function presented in (Miech et al. 2020)⁶. The proposal features are of dimension 1024 and only used for ASRL-QA extracted using FasterRCNN (Ren et al. 2015) trained on Visual Genome (Krishna et al. 2016).

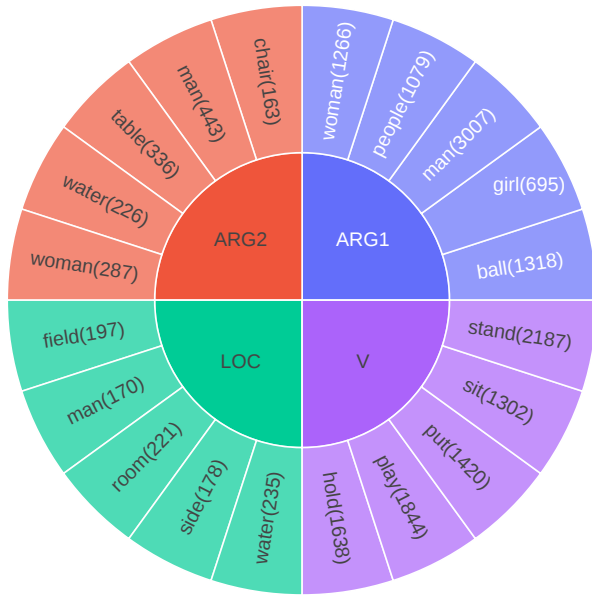
⁴github url: <https://github.com/tylin/coco-caption>

⁵github url: https://github.com/Tiiiger/bert_score

⁶<https://github.com/antoine77340/S3D-HowTo100M>



(a) Top-5 lemmatized nouns or verbs for the considered semantic roles in ASRL-QA



(b) Top-5 lemmatized nouns or verbs for the considered semantic roles in Charades-SRL-QA

Figure 1: Lemma Distribution for both ASRL-QA and Charades-SRL-QA. The number of instances across the whole dataset are given in the parenthesis of each lemmatized noun or verb.

For all cases, we report the output dimension of MLP. Unless otherwise stated, MLP is followed by ReLU activation.

Decoder: The decoder uses an input of $T \times 512$ (where T refers to the length of the input embedding). Note that for Lang-QAP, T is same as sequence length of the query, for

BUTD-QAP $T=1$, for VOG-QAP, T is number of SRLs * number of segment features. For MTX-QAP, T is sequence length of query + number of segment features. To generate output sequences, we use the usual beam-search with a beam-size of 2, with a temperature of 1.0.

Encoder: Encoder differs based on the specific model. All encoders are transformer based using 8 attention heads and 3 layers unless otherwise mentioned.

Lang-QAP: The language encoder uses 3 encoding layers, with 8 attention heads each. The embedding layer uses a dimension of 512.

BUTD-QAP: We use the same language query, with and pre-pend a $[CLS]$ token. The embedding of the $[CLS]$ token serves as the language embedding, and is passed through a MLP of dimension 512. The language encoder is the same as Lang-QAP. The segment features are passed through MLP of dimension 512. If proposal features are used, they are passed through a separate MLP of dimension 512. The language embedding (also of dimension 512) is used to compute attention score with the visual features, and finally obtain an attended visual feature. These attended visual features are concatenated with the language embedding along the last axis, and then passed to the decoder.

VOG-QAP: We use the same language encoder, but further use the SRL phrase start and end-points for the phrase encoder. The phrase encoder uses these start and end points to gather the language embeddings corresponding to these start and end points, concatenate them (dimension $512+512=1024$) and use MLP with dimension 512. This gives an output of the phrase encoder of size number of SRLs * 512. The phrase encoded query is then concatenated with all the segment features and passed through a MLP. Finally a multi-modal transformer encoder is applied over the phrase encoded input, and is passed to the language decoder.

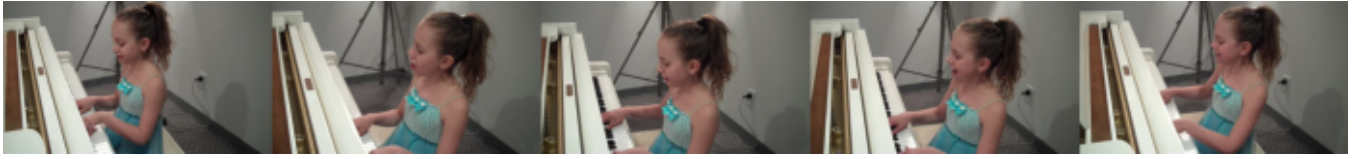
MTX-QAP: We collate all the language tokens (passed through embedding layer) as well as segment features passed through MLP, to get all features of dimension 512. A transformer based encoder is applied on these features, and the output is passed to the decoder.

Training: We train using standard cross-entropy loss (errata in main text which states smooth cross entropy). The decoder is trained using teacher forcing. All models are trained for 10 epochs with batch size of 32. On a TitanX, each epoch takes around 30 – 40 mins.

C Visualization

We visualize the model outputs on ASRL-QA in Figure 2 (a), (b), Figure 3 (a), (b) and Figure 4. For each case, we show the considered input in the first row, and the contrastive sample in the second row. Each row contains 5 frames uniformly sampled from the video segment to be representative of the content observed by the model. For every query, we show the ground-truth answer and the outputs from Lang-QAP, BUTD-QAP, VOG-QAP and MTX-QAP.

Overall, we often find Lang-QAP suggesting very probable answers, but as expected they are not grounded in the video. As a result, in either of the original sample or the contrastive sample, it performs poorly.



Query: <Q-ARG0> play the song on the piano

Target Answer: A little girl

Lang-QAP: The man

BUTD-QAP: A young child

VOG-QAP: A woman

MTX-QAP: The woman



Query: <Q-ARG0> playing a song

Target Answer: A man wearing a hat

Lang-QAP: A woman

BUTD-QAP: A man

VOG-QAP: A man wearing a hat

MTX-QAP: A man

(a) Query of type ARG0



Query: A man <Q-V> a skateboard

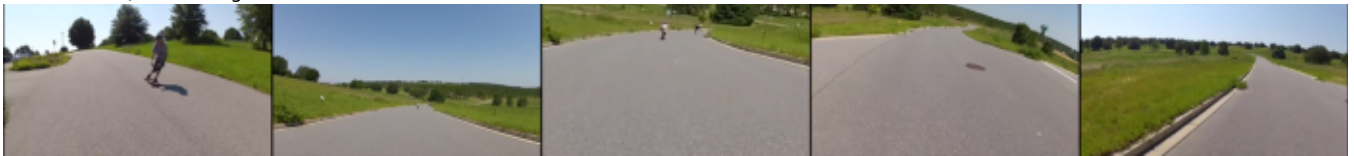
Target Answer: holding

Lang-QAP: riding

BUTD-QAP: picks

VOG-QAP: holding

MTX-QAP: holding



Query: Men <Q-V> skateboards

Target Answer: riding

Lang-QAP: riding

BUTD-QAP: riding

VOG-QAP: riding

MTX-QAP: riding

(b) Query of type V

Figure 2: Queries of Type ARG0 and V on ASRL-QA



Query: People hit <Q-ARG1>

Target Answer: a pinata

Lang-QAP: the ball

BUTD-QAP: the pinata

VOG-QAP: the pinata

MTX-QAP: the pinata



Query: The people hit <Q-ARG1>

Target Answer: the ball

Lang-QAP: the ball

BUTD-QAP: the ball

VOG-QAP: the ball

MTX-QAP: the ball

(a) Query of type ARG1



Query: A man sitting <Q-ARG2>

Target Answer: behind a drum kit

Lang-QAP: on a bed

BUTD-QAP: on a drum set

VOG-QAP: behind a drum set

MTX-QAP: in front of a drum set



Query: A man sits <Q-ARG2> next to a baby

Target Answer: on a playground swing

Lang-QAP: on a bed

BUTD-QAP: on the ground

VOG-QAP: on a swing

MTX-QAP: on a swing

(b) Query of type ARG2

Figure 3: Queries of Type ARG1 and ARG2 on ASRL-QA



Query: A lady washing clothes <Q-ARGM-LOC>

Target Answer: in a bucket

Lang-QAP: in a sink

BUTD-QAP: in a bowl

VOG-QAP: in a bucket

MTX-QAP: in the water



Query: People washing their clothes <Q-ARGM-LOC>

Target Answer: in a river

Lang-QAP: in a sink

BUTD-QAP: in a lake

VOG-QAP: on a river

MTX-QAP: in the water

Figure 4: Queries of Type ARGM-LOC on ASRL-QA

References

- Anderson, P.; He, X.; Buehler, C.; Teney, D.; Johnson, M.; Gould, S.; and Zhang, L. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition* 6077–6086.
- Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, 86–90.
- Banerjee, S.; and Lavie, A. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *IEEevaluation@ACL*.
- Bonial, C.; Hwang, J.; Bonn, J.; Conger, K.; Babko-Malaya, O.; and Palmer, M. 2012. English propbank annotation guidelines. *Center for Computational Language and Education Research Institute of Cognitive Science University of Colorado at Boulder* 48.
- Chen, X.; Fang, H.; Lin, T.-Y.; Vedantam, R.; Gupta, S.; Dollár, P.; and Zitnick, C. L. 2015. Microsoft COCO Captions: Data Collection and Evaluation Server. *ArXiv abs/1504.00325*.
- Denkowski, M.; and Lavie, A. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*.
- Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*.
- Gardner, M.; Grus, J.; Neumann, M.; Tafjord, O.; Dasigi, P.; Liu, N. F.; Peters, M.; Schmitz, M.; and Zettlemoyer, L. S. 2017. AllenNLP: A Deep Semantic Natural Language Processing Platform. In *Arxiv*.
- Goyal, Y.; Khot, T.; Summers-Stay, D.; Batra, D.; and Parikh, D. 2017. Making the V in VQA Matter: Elevating the Role of Image Understanding in Visual Question Answering. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 6325–6334.
- Gupta, S.; and Malik, J. 2015. Visual Semantic Role Labeling. *ArXiv abs/1505.04474*.
- Heilman, M.; and Smith, N. A. 2009. Question generation via overgenerating transformations and ranking. Technical report, Carnegie-Mellon Univ Pittsburgh pa language technologies insT.

- Hudson, D. A.; and Manning, C. D. 2019. GQA: a new dataset for compositional question answering over real-world images. *ArXiv abs/1902.09506*.
- Jang, Y.; Song, Y.; Yu, Y.; Kim, Y.; and Kim, G. 2017. TGIF-QA: Toward Spatio-Temporal Reasoning in Visual Question Answering. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 1359–1367.
- Jasani, B.; Girdhar, R.; and Ramanan, D. 2019. Are we Asking the Right Questions in MovieQA? *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* 1879–1882.
- Joshi, M.; Chen, D.; Liu, Y.; Weld, D. S.; Zettlemoyer, L.; and Levy, O. 2019. SpanBERT: Improving Pre-training by Representing and Predicting Spans. *Transactions of the Association for Computational Linguistics* 8: 64–77.
- Kingsbury, P. R.; and Palmer, M. 2002. From TreeBank to PropBank. In *LREC*, 1989–1993. Citeseer.
- Krishna, R.; Zhu, Y.; Groth, O.; Johnson, J.; Hata, K.; Kravitz, J.; Chen, S.; Kalantidis, Y.; Li, L.; Shamma, D. A.; Bernstein, M. S.; and Fei-Fei, L. 2016. Visual Genome: Connecting Language and Vision Using Crowdsourced Dense Image Annotations. *International Journal of Computer Vision* 123: 32–73.
- Lee, K.; He, L.; Lewis, M.; and Zettlemoyer, L. 2017. End-to-end Neural Coreference Resolution. In *EMNLP*.
- Lei, J.; Yu, L.; Bansal, M.; and Berg, T. 2018. TVQA: Localized, Compositional Video Question Answering. In *EMNLP*.
- Li, M.; Zareian, A.; Zeng, Q.; Whitehead, S.; Lu, D.; Zhong, J.; and Chang, S.-F. 2020. Cross-media Structured Common Space for Multimedia Event Extraction. In *ACL*.
- Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *ACL 2004*.
- Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *ArXiv abs/1907.11692*.
- Lu, J.; Batra, D.; Parikh, D.; and Lee, S. 2019. ViLBERT: Pretraining Task-Agnostic Visiolinguistic Representations for Vision-and-Language Tasks. *ArXiv abs/1908.02265*.
- Maharaj, T.; Ballas, N.; Rohrbach, A.; Courville, A. C.; and Pal, C. 2017. A Dataset and Exploration of Models for Understanding Video Data through Fill-in-the-Blank Question-Answering. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 7359–7368.
- Miech, A.; Alayrac, J.-B.; Smaira, L.; Laptev, I.; Sivic, J.; and Zisserman, A. 2020. End-to-End Learning of Visual Representations From Uncurated Instructional Videos. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 9876–9886.
- Miech, A.; Zhukov, D.; Alayrac, J.-B.; Tapaswi, M.; Laptev, I.; and Sivic, J. 2019. HowTo100M: Learning a Text-Video Embedding by Watching Hundred Million Narrated Video Clips. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* 2630–2640.
- ning Yang, J.; Zhu, Y.; Wang, Y.; Yi, R.; Zadeh, A.; and Morency, L.-P. 2020. What Gives the Answer Away? Question Answering Bias Analysis on Video QA Datasets. *ArXiv abs/2007.03626*.
- Ott, M.; Edunov, S.; Baevski, A.; Fan, A.; Gross, S.; Ng, N.; Grangier, D.; and Auli, M. 2019. fairseq: A Fast, Extensible Toolkit for Sequence Modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.
- Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *ACL*.
- Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Kopf, A.; Yang, E.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems* 32, 8024–8035. Curran Associates, Inc. URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- Pennington, J.; Socher, R.; and Manning, C. D. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*.
- Pradhan, S.; Moschitti, A.; Xue, N.; Ng, H. T.; Björkelund, A.; Uryupina, O.; Zhang, Y.; and Zhong, Z. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, 143–152.
- Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39: 1137–1149.
- Sadhu, A.; Chen, K.; and Nevatia, R. 2020. Video Object Grounding Using Semantic Roles in Language Description. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 10414–10424.
- Shi, P.; and Lin, J. 2019. Simple BERT Models for Relation Extraction and Semantic Role Labeling. *arXiv preprint arXiv:1904.05255*.
- Sigurdsson, G. A.; Varol, G.; Wang, X.; Farhadi, A.; Laptev, I.; and Gupta, A. 2016. Hollywood in Homes: Crowdsourcing Data Collection for Activity Understanding. In *ECCV*.
- Silberer, C.; and Pinkal, M. 2018. Grounding Semantic Roles in Images. In *EMNLP*.
- Strubell, E.; Verga, P.; Andor, D.; Weiss, D.; and McCallum, A. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. *ArXiv abs/1804.08199*.
- Tapaswi, M.; Zhu, Y.; Stiefelshagen, R.; Torralba, A.; Urtasun, R.; and Fidler, S. 2016. MovieQA: Understanding Stories in Movies through Question-Answering. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 4631–4640.

- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. *ArXiv abs/1706.03762*.
- Vedantam, R.; Zitnick, C. L.; and Parikh, D. 2015. CIDEr: Consensus-based image description evaluation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 4566–4575.
- Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Gool, L. 2016. Temporal Segment Networks: Towards Good Practices for Deep Action Recognition. In *ECCV*.
- Xie, S.; Sun, C.; Huang, J.; Tu, Z.; and Murphy, K. 2018. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. In *ECCV*.
- Xu, D.; Zhao, Z.; Xiao, J.; Wu, F.; Zhang, H.; He, X.; and Zhuang, Y. 2017. Video Question Answering via Gradually Refined Attention over Appearance and Motion. In *ACM Multimedia*.
- Yatskar, M.; Zettlemoyer, L.; and Farhadi, A. 2016. Situation Recognition: Visual Semantic Role Labeling for Image Understanding. In *Conference on Computer Vision and Pattern Recognition*.
- Yu, Z.; Xu, D.; Yu, J.; Yu, T.; Zhao, Z.; Zhuang, Y.; and Tao, D. 2019. ActivityNet-QA: A Dataset for Understanding Complex Web Videos via Question Answering. In *AAAI*, 9127–9134.
- Zellers, R.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. From Recognition to Cognition: Visual Commonsense Reasoning. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 6713–6724.
- Zeng, K.-H.; Chen, T.-H.; Chuang, C.-Y.; Liao, Y.-H.; Niebles, J. C.; and Sun, M. 2017. Leveraging Video Descriptions to Learn Video Question Answering. In *AAAI*.
- Zhang, T.; Kishore, V.; Wu, F.; Weinberger, K. Q.; and Artzi, Y. 2020. BERTScore: Evaluating Text Generation with BERT. *ArXiv abs/1904.09675*.
- Zhou, L.; Kalantidis, Y.; Chen, X.; Corso, J. J.; and Rohrbach, M. 2019. Grounded Video Description. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* 6571–6580.
- Zhu, L.; and Yang, Y. 2020. ActBERT: Learning Global-Local Video-Text Representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.