Improving Representations for Video Question Answering

Andrew Lyubovsky and Nour Jedidi
Language Technologies Institute
Carnegie Mellon University
{alyubovs, njedidi}@cs.cmu.edu
Team Name: Norew

Abstract

Video question answering (VideoQA) is a rapidly evolving topic that bridges the intersection between visual, audio and text domains with an added complexity of time. Though there has been rapid improvement in vision-language models, VideoQA still underperforms humans baselines. To address this, our paper studies methods for improving representations VideoQA. More specifically, our research focuses on the following questions: 1) How useful are video captions in VideoQA tasks?; and 2) How can we leverage the knowledge of frames being close together in time to improve the representations? Through experiments over AGQA, we show that with well generated captions, our proposed Video Captioning for VideoQA (VC-VQA) approach is beneficial to downstream performance over AGQA and improves upon previous baselines. In addition, leveraging similar representations for clips that occur close together in a video improves performance for questions that require knowledge of the timing of an action.

1 Introduction

Video question answering (VideoQA) is a rapidly evolving topic that bridges the intersection between visual, audio and text domains with an added complexity of time. VideoQA is interesting in the aspect of combining dense image representations with a temporal dimension, in which the images shift slightly from one instance to the next. In recent years there have been major breakthroughs in individual modalities with popular models such as AlexNET (Krizhevsky et al., 2012) in the visual domain and BERT (Devlin et al., 2019) in the text domain. Similarly, there have been vast improvements in the VideoQA task, however, current models still significantly underperform human comparisons (Grunde-McLaughlin et al., 2021) and is still largely unexplored (Zhong et al., 2022).

At a high level, VideoQA seeks to, given a question, q and a video clip v, predict the correct answer, a^* (Zhong et al., 2022). Traditional approaches to VideoQA consist of various components which encode the video, the question, and the cross-modal interaction of the two (Zhong et al., 2022). However, given the success of transformer models across text-based question answering tasks, the use of textual descriptions of videos has not been explored in depth in the context of VideoQA (Patel et al., 2021). This angle for VideoQA is particularly interesting as it allows us to convert the VideoQA task to a text-based QA task, where pre-trained language transformer models excel (i.e., (Devlin et al., 2019), (Raffel et al., 2020), (Lewis et al., 2019)). To this end, our paper proposes Video Captioning for VideoQA (VC-VQA), a two-step approach which first generates captions for videos and then makes use of a pre-trained language models for question answering.

We study VC-VQA across two setups: 1) There is access to human written train and test video captions for our question-answering model *Perfect world*; and 2) The question answering model has access to human written captions for training, but need to we need to generate captions for test videos *Semi-Perfect World.*

In addition to VC-VQA, we also explore the impact of temporal reasoning on VideoQA. We hypothesize that the structure of video data can be leveraged to create a computational process with stronger parallels to human cognitive processes. Recent work in visual question answering has explored aspects of reasoning, in which multiple parts of an image have to be properly understood in order to answer questions (Amizadeh et al., 2020). Thus, we believe that the temporal aspect of frames can be better incorporated to improve upon that. Broadly speaking, from human reasoning, we can observe that time is a necessary aspect that allows human reasoning to occur. There is also theoretical concepts of *System 1* and *System 2* reasoning, where

System 1 reasoning refers to fast cognitive abilities such as object recognition, and *System 2* refers to cognitive processes that are slower (Susskind et al., 2021). While there has been extensive research on using time dependent modalities such as audio and video data, there has been little exploration of how this effects the representations within the models. As such, we aim to explore how the temporal effects that occur in VideoQA.

In summary, we make the following contributions: 1) We study if textual descriptions of videos can be beneficial to downstream performance over AGQA and improve upon previous baselines. 2) We study the effectiveness of leveraging similar representations that occur close together in a video for question answering.

Our paper is organized as follows. In section 2 we go over related work. In section 3, we overview the AGQA dataset. In section 4, we describe the details of our proposed approach. In section 5, we discuss our experimental setup. Section 6 discusses the experimental results. Section 6 concludes our study.

2 Related Work

Our work is related to video question answering, text-based question answering, automatic video captioning, and temporal patterns.

VideoQA In the recent years, there have been an increasing number of models and datasets that have been aimed at addressing this task. Multiple datasets have been proposed, where often videos are either taken from previous datasets (Ji et al., 2020), or scraped from online sources (Li et al., 2020; Jang et al.; Xu et al., 2016). Questions are then either manually created or automatically generated based on previous labels . Current state of the art models are trained over large datasets such as HowTo100M, with up to 100M video question answer pairs (Miech et al., 2019). However, a critique of them is that they can ask about information that is present outside of a video (Grunde-McLaughlin et al., 2021). As such, our work focuses on the AGQA dataset, where question require an understanding of the videos to be answered (Grunde-McLaughlin et al., 2021).

HCRN is a popular videoQA model that has multiple Conditional Relational Networks (CRN)s. By combining these modules, it has achieved state of the art performance performance on multiple benchmarks including TGIF-QA and MSRVTT-QA datasets (Le et al., 2020). The modules condition image features on questions and motion features (I3D video features) to extract more relevant information from the input features. The model stacks one such layer on top of another, where the later layer predicts the answer to a question. This way, contextual information form the question and motion features can be combined with appearance features from the videos in multiple places within the model (Le et al., 2020). A skeleton of the model can be observed in Fig. 1.

Video Captioning The goal of video captioning is to generate natural language descriptions which describe the content of a video (Aafaq et al., 2019). Video Captioning has many interesting downstream applications to tasks such as video retrieval or video recommendation.

In our work, video captioning is used as a method to leverage text modality for VideoQA. More specifically we use a video captioning model to directly generate a description of a video for input into text-based question answering system. The work in video captioning most related to ours is BMT (Iashin and Rahtu, 2020) and ClipCap (Mokady et al., 2021).

BMT is a dense video captioning model which utilizes both audio and visual features to detects events in a video then describes them (Krishna et al., 2017). BMT model consists of two modules: event proposal generation module and a captioning module. The event proposal generation module is a bi-modal transformer model that uses video and audio features to generate a set of proposals (Iashin and Rahtu, 2020) The captioning module is a bi-modal encoder-decoder transformer model which outputs a caption for each "proposal" from the event proposal generation module. The videos are encoded using pre-trained I3D features (Carreira and Zisserman, 2017) and pre-trained VGGish features (Hershey et al., 2017).

ClipCap is an image captioning model which leverages CLIP encodings (Radford et al., 2021) as a prefix for generating captions with GPT-2 through the use of a mapping network (Mokady et al., 2021). We use a similar approach to expand upon this for video captioning by leveraging video encodings rather than CLIP image encodings.

Text-based Question Answering Our work is most related to the closed-book question answering task, in which given a question and a context paragraph, models are trained to predict the span of text

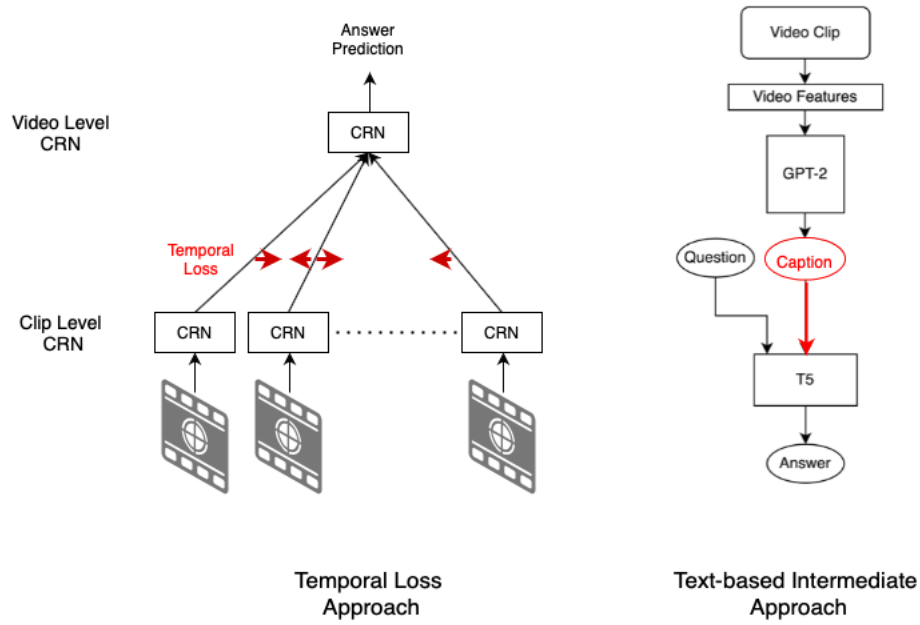

Figure 1: The figure highlights our main contributions. We propose a method to enforce more consistent representations in video, and utilize the text modality through captions for the video QA task.

which contain the answer (Roberts et al., 2020).

In particular, our approach leverages out-of-the-box pre-trained language models (PLMs) that follow the standard pre-train then fine-tune recipe. Traditionally for question-answering, PLMs are first pre-trained on large text corpora with tasks such as masked-language modeling followed by fine-tuning over (question, context, answer) triples. Popular PLM architectures consist of *encoder* models similar to BERT (Devlin et al., 2019) which are trained to predict the span of tokens which contain the answer or *encoder-decoder* models like T5 (Raffel et al., 2020) or BART (Lewis et al., 2019) which directly generate the answer.

Temporal Patterns Recently, there has been increased attention devoted to temporal relations that occur in VideoQA. Recently, Spatio-temporal Question Answering datasets have become more popular, evaluating the ability of models to answer questions about both spacial relationships and temporal relationships that are present within a video. These datasets focus on questions that require knowledge from different time steps within a video, evaluating the ability of a model to collect information across different time horizons. Such questions ask about recognizing actions and temporal relationships such as whether something happened before or after smoothing else, or whether one action was longer than another action. While there have been many related datasets that aim to evaluate temporal relationships, some approaches also aim to specifically temporal information.

Goyal et al. (2020) apply the concept of temporal coherence to the image classification task, where they attend more to frames that do not change, using the by looking at the similarity of temporaly proximal frames. Attention for a frame is scaled by the similarity between the frame's attention vector and the attention vectors of its neighbors. This makes the shift in attention smooth from one frame to the next, making it more consistent.

Yu et al. (2021) force similarity amongst similar video representations that have a slight temporal offset. That is, they sample frames from a video, and compare that sample with a similar sample taken with a slight delay. They compare these representation, giving more weight to those that are similar. As representations that appear at neighboring times and are similar suggest more reliable information, they are given more weight when making predictions for the answer.

These prior works are able to leverage information about similar representations from neighboring time steps to improve video question answering. As such, we believe that we can enforce similar representations to improve the task of videoqa on the AGQA dataset.

3 AGQA Dataset (Grunde-McLaughlin et al., 2021)

The AGQA dataset is a question answering dataset based on the action genome with questions that offers insight about the 'reasoning' that a model is able to perform.

The videos are collected from the Charades dataset (Sigurdsson et al., 2016), in which 9848 videos of indoor activities were collected. People were asked to do a specific sequence of actions, and recorded videos of themselves doing the actions. For example one video is "A person [taking] a drink of water, then [closing] a cabinet door, then [walking] out of the room". The videos have an average length of 30 seconds, and the videos are limited to 157 action classes (Sigurdsson et al., 2016). Each video is annotated with the multiple forms for textual description. The first is a *script*, which is the task the person was provided with and requested to act out in the creation of the video. The second is a *description*, which is a sentence description of the video written by workers who watched the video and described what they saw.

Next, the Action Genome created temporal scene graphs based on the videos, where temporal scene graphs are generated by annotating five frames through the duration of each action within a video (Ji et al., 2020). Based on the scene graphs, the AGQA dataset creates 3.9 million question pairs, with 174 unique answers, where the question types and answers are relatively balanced. They do so by creating question templates based on the temporal scene graphs, and a program that generates the answer to the question based on the scene graph (Grunde-McLaughlin et al., 2021).

The questions in AGQA focus on video understanding, requiring an the ability to recognize concepts that occur in the video, actions, and relationships. Moreso, the questions often require comparisons between actions and objects, asking about which action was done first, or which action was done for a longer period of time. For each question, they differentiate between the question reasoning type, semantics, and structure. Question semantics refers to what the question is asking about, reasoning type refers to the concepts that need to be used to answer a question, and question structure refers to the way in which they ask about the concept. This allows the models to be evaluated on the ability to pick up different concepts and answer different question types, providing a more informative evaluation metric about different reasoning abilities.

4 Approaches

In this section, we first outline our proposed Video Captioning for VideoQA (VC-VQA) approach. Then, we describe our methodology for applying a temporal loss.

4.1 VC-VQA

Unlike previous works, which directly utilize the video clip as input to a question answering architecture, our proposed model seeks to leverage textual representations in order to improve the video question answering task. Current language models such as T5 (Raffel et al., 2020) or BERT (Devlin et al., 2019) are pre-trained on large text corpora, and as a result leverage knowledge extracted from textual representations. Our proposed method seeks to gain an understanding the role that textual representations can play in the context of video question answering. First, we evaluate the ability of text based models to answer questions based off of human curated captions of the video. Then we explore off-the-shelf video captioning models to generate useful captions for the QA downstream task. Lastly, we evaluate the ability to generate captions using models fine-tuned on captions for the downsteram task of VideoQA. Overall, VC-VQA consists of a two-step pipeline that first generates captions for videos and then makes use of a pre-trained language model for question answering as seen in Fig 1.

Video Captioning Module At a high level, the video captioning component of VC-VQA can be any off-the-shelf model which takes as input a video clip, and outputs a textual description of the video. However, to leverage the benefits of pre-training in language generation models, we train a GPT-2 (Radford et al., 2019) model for video captioning. This approach is similar to a clip cap model proposed by (Mokady et al., 2021). There, image features were embeded into the same space as textual embeddings, and then passed into GPT-2. In our case GPT-2 is fine-tuned to generate captions given weights from image features.

Question Answering Module For the question answering model, we fine tune a text-to-text transformer (T5) model for question answering. Given a dataset of (Question, Caption, Answer) triples, T5 takes as input a concatenation of the question

and caption and outputs an answer. The following is the input to T5:

$$\text{Question: } q \text{ Context: } c \quad (1)$$

where q is the question and c is the video caption.

4.2 Temporal Reasoning

In order to leverage temporal coherence (the concept that neighboring representations should be similar) as a way of regularizing to HCRN, we aim penalize the change in representations that occurs within the middle layers of the network. This way, the network will be able to learn representations that remain consistent thought a duration of a video. For example, if a person is doing a continuous action, that would get stored within the representation instead of individual positions through which a person transitions. We hope that the penalty will be weak enough for the representation to change throughout the clips of the video to reflect the specifics of each frame, yet strong enough that some information that is consistent throughout multiple consecutive frames is extracted. This way, the representation is able to better capture the events, while preserving details from each frame. Since all of the videos within the AGQA data set are single clip videos, we do not worry about sharp transitions, and while multiple videos have fast transitions, we hope that the metric will be robust to these transitions. We apply this penalty to vectors that are returned from the first layer of CRN modules.

5 Experimental Setup

In this section we first overview the research questions that are aimed to be answered by our experiments. Then we discuss data and implementation details for VC-VQA and incorporation of temporal loss.

5.1 Research Questions

RQ1: How useful are video captions in VideoQA tasks? If so, can video captioning models generate captions that can have similar success to human-generated captions?

RQ2: How can we leverage the knowledge of frames being close together in time to improve the representations?

5.2 Model Training

T5 We fine-tune a T5-base model (220M parameters) for approximately 6000 steps (2 epochs) using the Adafactor optimizer with a learning rate of 3e-4, batch size of 256, and weight decay of 5e-5. We used a maximum of 512 input tokens and 64 output tokens. We select the epoch with the lowest evaluation loss as the final model for inference. T5 was implemented using the huggingface transformers library (Wolf et al., 2019). Training and inference was done on 1 NVIDIA GeForce 2080 Ti GPU and takes approximately 7 hours.

GPT-2 We fine-tune a GPT-2 model (117M parameters) for video captioning. Given GPT-2 is a transformer model that is used to generate text, we pose the question of whether video features can prompt GPT-2 to generate useful video captions. Given the appearance features have dimensions of 2048 – and GPT-2 has dimensions of 768 – we slice the appearance features at each frame into three consecutive inputs for GPT-2. We also experiment with an alternative approach to project the 2048 dim image feature into 768 dimensions. Next, we unfreeze the first 3 layers of GPT-2, and keep the rest of the layers frozen. This way, the output does not overfit and is able to pick up more nuances from the input visual features. We train this for 10 epochs using the huggingface transformers library (Wolf et al., 2019) on a 1 NVIDIA GeForce 2080 Ti GPU and takes approximately 7 hours and takes approximately 10 hours.

BMT (Iashin and Rahtu, 2020) For a video captioning baseline to compare with GPT-2, we fine-tune BMT (60M parameters) over the AGQA training videos. Although BMT utilizes both audio and visual features, we omit the audio features in our implementation, as the majority of videos in Charades do not include audio. In addition, given we have only one caption per video, we freeze the proposal generator, and only train the captioning module. We train the BMT captioning module for 30 epochs with the default parameters from (Iashin and Rahtu, 2020) and select the epoch with the lowest evaluation loss. Training and inference was done on 1 NVIDIA GeForce 2080 Ti GPU and takes approximately 2 hours.

5.3 Temporal Loss

Our last approach aims to leverage information about proximal representation to improve the performance of HCRN (Le et al., 2020). HCRN is

a 500 MB model, and takes 48 GPU hours to train on the AGQA dataset. To make representations that come consecutively more similar, we experiment with different loss functions that penalize side by side representations changing. To calculate the loss, we apply the frobenius, neuclear, and p2 matrix norms. The Frobenius (p1) matrix norm penalizes all changes in representations equally. This way, any change in the representation gets penalized. On the other hand, the Nuclear Matrix norm penalizes the sum of the singular values for the matrix. That is, it penalizes the total amount by which a matrix stretches, resulting in larger penalties over dimensions that stretch further. Lastly, the p2 norm penalizes all values where that change using the sum of squares of all of the elements (Horn and Johnson, 2012). This penalizes large changes in values from one time step to the next. When combining the loss from changing representations with the cross entropy loss from answer predictions using different weights for the loss. This is expressed as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{crossentropy} + \alpha\mathcal{L}_{temporal}$$

In order to evalute the performance we first train the model with different model weights and regularizations from scratch, sampling from weigts of 1e-2 to 1e-5 and training for 2 epochs each. Next, we train a model using no regularization for 7 epochs, and then train three models models with an p2 norm for 6500 more iterations (.1 epoch). We evaluate their performance on the different AGQA splits, and use the 1 sample t test to compare the final accuracy distributions with the original model performance.

Having set up the experiments, we disucss the results next.

6 Results and Analysis

6.1 RQ1: Do video-captioning models help VideoQA?

Table 2 reports the accuracy of various setups of VC-VQA comparatively to reported baselines from AGQA (Grunde-McLaughlin et al., 2021). Comparing VC-VQA to the reported model baselines, it can be seen that performance can be improved by 20% overall through the use of human written captions. These findings confirm our hypothesis that given we provide a question-answering model with well written descriptions of videos (captions), performance can improve for VideoQA tasks.

Comparison of VC-VQA with human written captions versus automatically generated captions shows that generated captions can still outperform previous VideoQA baselines over AGQA, although there is still room for improvement compared to human written captions. Comparing VC-VQA over generated captions to the baseline shows that the improvement of VC-VQA is most evident over open-answer questions, which have many possible answers (Grunde-McLaughlin et al., 2021). This shows the benefit of treating VideoQA as a text question answering task, as the question-answering module of VC-VQA is better able to utilize textual representations of videos versus baselines that directly use video features. In addition, we hypothesize this is because our VC-VQA question answering module is able to leverage the knowledge extracted from pre-trained language model textual representations, which excel in text-based question answering.

To gain a better understanding of the drop in performance compared to performance over human written captions, Table 3 shows examples of captions generated by humans, BMT, and GPT-2. Interestingly, BMT is able to capture the action the person is performing, but is struggling to classify the object from the video. (I.e., in row 1, BMT is able to notice the person picks something up but predicts it is a "bag" versus a "pillow"). On the other hand, GPT-2 does a much better job of capturing the objects and actions present in the video. Row 3 shows an interesting example of where the human caption doesn't explicitly mention the object being held by the person, but both BMT and GPT-2 make predictions on what the object is.

To quantify this analysis, we compute the METEOR (Banerjee and Lavie, 2005) score of the generated captions to the human written captions. As we can see GPT-2 captions has a four point improvement in METEOR, which further confirms our findings that GPT-2 better captures unigram precision and recall compared to the human description versus BMT.

6.2 RQ2: Leveraging knowledge of frames to improve video representations

In order to better understand relationships between different representations within the HCRN model, we calculate the euclidean distances amongst video appearance features, and amongst the activations in the HCRN model.

		weight of loss (α)			
		.01	$1e^{-3}$	$1e^{-4}$	$1e^{-5}$
Norm	forb (p1)	44.77	44.88	46.99	47.14
	nuc	44.50	46.60	47.05	47.26
	p2	46.3	47.79	46.79	47.21

Table 1: The table presents the performance of HCRN when different forms of regularization are applied to the model.

	AGQA		
Method	Binary	Open	All
Baselines			
Human	86.65	83.53	86.02
PSAC	54.19	27.20	40.40
HME	59.77	36.23	47.74
HCRN	58.11	37.18	47.42
Human Written Captions			
VC-VQA	67.85	67.20	67.26
Automatically Generated Captions			
VC-VQA (w/GPT-2)	58.90	52.69	55.26
VC-VQA (w/BMT)	56.75	39.66	47.58

Table 2: Results on AGQA

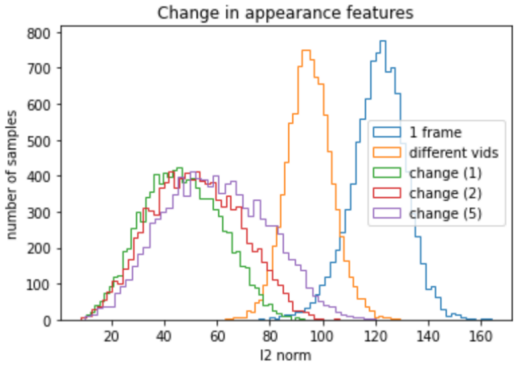

Figure 2: The figure shows the how the representations change from one clip to the next.

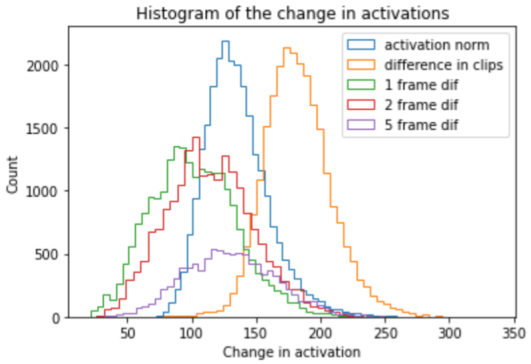

Figure 3: The figure shows how the HCRN activations change from one HCRN representation to the next.

From Figures 2 and 3, we can visualize how much the representations change as a function of time. We observe that for appearance (ResNet) features that were fed into the model, there is minimal variation between how much features change between one clip and five clips. However, looking at the change in representations in Fig. 3, we can see that there is a more noticeable change in the change of representation features, as the average grows from an average p2 norm of 100 for image features 1 segment away to 130 when the image features are 5 segments away. We can also observe that the activation norm is much smaller for activations compared to appearance features. This suggests that representations change more significantly within the HCRN model compared to the change that occurs in the original image features. Using this technique, we can note that HCRN is able to extract meaningful information from image features. Similarly, the figures suggest the lack of granularity between the image features/representations as there is little observed change between one and 5 video frames.

Next, we evaluated how adding a loss to how much a model changes can affect the performance of the model. From Table 1, we observe that when

Ground Truth	BMT	GPT-2
A person looks under a mattress and pats the bed. This person picks up a pillow, and throws it on the bed	A person is standing on the stairs holding a bag and a towel . the person then picks up a bag and throws it on the floor .	A person awakens on the edge of the living room. The person picks up a pillow and throws it on the bed.
A person is walking into a closet and begins looking through things, they then take a picture from a shelf and leave.	A person is looking out the window, they then go to the window and look out the window.	A person walks out of the closet.
A person walks down the stairs holding something , and then opens a door and walks through it.	A person is walking up the stairs and looking at a phone while another person is walking down the stairs .	A person is walking down the stairs holding a camera. The person then opens a door and walks through it.

Table 3: Examples of generated captions versus human written captions

Method	METEOR
BMT	17.01
GPT-2	21.95

Table 4: METEOR score for generated captions over Charades test set

a larger weight that gets applied to the time-based loss, the performance of the model decreases. This remains consistent for the different kinds of norms that get applied. We can also observe that in most cases, the p2 norm outperforms the Nuclear and Frobenius norms, which suggests that rather than the change of the whole representation (which is reflected in the nuclear norm), the change in individual components of the representation carries more meaningful information.

Lastly, we add the p2 norm with the loss, and train the model for 6500 iterations (.1 epoch) to evaluate how the norm affects the questions that the model can answer. We can observe a slight improvement in performance, however the p value still remains relatively large (.27). We can observe the most improvement in the questions where the structure requires a comparison between two actions (1.3 % improvement with a p value of .06), and when the question semantics ask about an Action (1.2 % improvement with a p value of .05). The results can be observed in Table 5.

Looking at Semantic: Action and Structural: Compare questions, we observe the types of questions that the baseline model gets wrong and the new model gets correct. We find that most of the questions are ones that ask to compare whether an action happened 'before or after' another action. While comparison and action questions had multiple types of comparisons and actions types (for example comparing the duration that something occurred, or comparing what object an action was applied to) most of the questions that the model was able to answer correctly asked about whether an event happened before or after a different event. These types of questions were both classified as Semantic: Action, and Structural: Compare. This suggests that adding a p2 temporal loss is able to improve the performance on questions that require a model to order events.

	Baseline	L2-loss	p-val
Total	47.2	47.62	.27
Binary	54.8	54.7	.88
Open	40.8	41.7	.17
Structural: Verify	67.3	67.4	.82
Structural: Query	40.8	41.7	.17
Structural: Choose	42.2	41.4	.29
Structural: Compare	54.9	55.6	.06
Semantic: Object	43.6	.44	.31
Semantic: Relation	64.4	64.4	.50
Semantic: Action	56.7	57.4	.05

Table 5: These experiments provide a comparison between the baseline model and a model that applied a loss to changing representations at different time steps

7 Discussion and Future Work

In this paper, we examine different methods for improving representations for VideoQA. VideoQA is particularly interesting as it combines dense image representations in a temporal sequence and has

multiple applications for information retrieval and recommendation systems.

To this end, we propose VC-VQA, a method that uses a two-step approach to first generates captions for videos and then makes use of a pre-trained language model for question answering. Through a variety of experiments with VC-VQA, we find that well generated captions provide an improvement to downstream performance over AGQA and significantly improves upon previous baselines, by up to 20%. We find this to also be the case with automatically generated captions, as we find up an 8% over previous AGQA baselines.

In addition, we study the effectiveness of leveraging similar representations that occur close together in a video for question answering. We find that these models are able to improve on the ordering of actions.

While VC-VQA outperforms previous baselines, we leave open the problem of zero-shot video captioning. Our approach assumes that – at minimum – the video collection contains human written captions for the training set. However, for cases in which a video collection has no human written captions, how VC-VQA performs remains an open question. In addition we chose to leverage temporal information on the HCRN model due to its successful previous performance on VideoQA, while these approaches might be more effectively applied to other model structures such as RNNs.

For future work, we will investigate how VC-VQA will perform in settings where training captions might not be available.

7.1 Social Implications

While AGQA splits the data based on question type, there are still many biases that remain in the dataset. First, the videos are collected by Amazon Turks, which is a select population that has access to recording devices and is interested in recording videos. As such, there could exist a gender and racial imbalance within the data set. As the videos are not categorized by race and gender, it becomes difficult to evaluate potential biases that may exist within the question answering system.

Another ethical implication of this work is the impact that it will have. Currently the most common application of such systems are in search engines, where videos can be retrieved to answer questions. As such, this direction of work has the most impact on individuals that are able to create high quality videos to get them retrieved and could have harmful effects to individuals who cannot do as such. Lastly, there can be possible unethical uses of such systems, for example, being used to track members of a certain group.

References

Nayyer Aafaq, Ajmal Mian, Wei Liu, Syed Zulqarnain Gilani, and Mubarak Shah. 2019. Video description: A survey of methods, datasets, and evaluation metrics. *ACM Computing Surveys (CSUR)*, 52(6):1–37.

Saeed Amizadeh, Hamid Palangi, Alex Polozov, Yichen Huang, and Kazuhito Koishida. 2020. Neuro-symbolic visual reasoning: Disentangling "Visual" from "Reasoning". In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 279–290. PMLR.

Satanjeev Banerjee and Alon Lavie. 2005. 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*, pages 65–72.

Joao Carreira and Andrew Zisserman. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. *ArXiv*, abs/1810.04805.

Palash Goyal, Saurabh Sahu, Shalini Ghosh, and Chul Lee. 2020. Exploiting temporal coherence for multi-modal video categorization. *CoRR*, abs/2002.03844.

Madeleine Grunde-McLaughlin, Ranjay Krishna, and Maneesh Agrawala. 2021. Agqa: A benchmark for compositional spatio-temporal reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. 2017. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE.

Roger A Horn and Charles R Johnson. 2012. *Matrix analysis*. Cambridge university press.

Vladimir Iashin and Esa Rahtu. 2020. A better use of audio-visual cues: Dense video captioning with bi-modal transformer. *arXiv preprint arXiv:2005.08271*.

Yunseok Jang, Yale Song, Chris Dongjoo Kim, Youngjae Yu, Youngjin Kim, and Gunhee Kim. Video Question Answering with Spatio-Temporal Reasoning.

Jingwei Ji, Ranjay Krishna, Li Fei-Fei, and Juan Carlos Niebles. 2020. Action genome: Actions as compositions of spatio-temporal scene graphs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10236–10247.

Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Thao Minh Le, Vuong Le, Svetha Venkatesh, and Truyen Tran. 2020. Hierarchical conditional relation networks for video question answering. *arXiv preprint arXiv:2002.10698*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

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

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

Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

Devshree Patel, Ratnam Parikh, and Yesha Shastri. 2021. Recent advances in video question answering: A review of datasets and methods. In *International Conference on Pattern Recognition*, pages 339–356. Springer.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the

limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? *arXiv preprint arXiv:2002.08910*.

Gunnar A. Sigurdsson, Gül Varol, X. Wang, Ali Farhadi, Ivan Laptev, and Abhinav Kumar Gupta. 2016. Hollywood in homes: Crowdsourcing data collection for activity understanding. *ArXiv*, abs/1604.01753.

Zachary Susskind, Bryce Arden, Lizy K. John, Patrick Stockton, and Eugene B. John. 2021. Neuro-symbolic AI: an emerging class of AI workloads and their characterization. *CoRR*, abs/2109.06133.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Jun Xu, Tao Mei, Ting Yao, and Yong Rui. 2016. Msr-vtt: A large video description dataset for bridging video and language. IEEE International Conference on Computer Vision and Pattern Recognition (CVPR).

Weijiang Yu, Haoteng Zheng, Mengfei Li, Lei Ji, Lijun Wu, Nong Xiao, and Nan Duan. 2021. Learning from inside: Self-driven siamese sampling and reasoning for video question answering. In *Advances in Neural Information Processing Systems*, volume 34, pages 26462–26474. Curran Associates, Inc.

Yaoyao Zhong, Wei Ji, Junbin Xiao, Yicong Li, Weihong Deng, and Tat-Seng Chua. 2022. Video question answering: Datasets, algorithms and challenges. *arXiv preprint arXiv:2203.01225*.