A Second Pair of Eyes - CS 231n Final Project

Arjun Karanam
Stanford University
akaranam@stanford.edu

Ronak Malde
Stanford University
rmalde@stanford.edu

Abstract

In this paper, we seek to tackle the project of Egocentric Video Question and Answer, with the goal of creating Augmented Reality systems that one could query for more information about the world around them. As opposed to traditional techniques of joint training across all modalities (in this case, egocentric video and language), we instead take the approach of composing multiple foundation models, using multi-modal informed captioning. This allows us to leverage the powerful priors in these foundational models while finetuning just one part, the Vision Language Model, with our egocentric data. We find that a pairing of PromptCap (a multimodal Vision Language Model) finetuned on data-augmented Egocentric videos + captions, composed with GPT3 yields the best results on the task set forth by the EgoVQA dataset. Using a separate Vision model to generate captions and GPT3 to answer the questions does not perform as well, demonstrating that there is still merit to jointly training a model with Egocentric Video and QA data in pursuit of the Egocentric Video Question and Answering task.

1. Introduction

Today, Augmented Reality hardware appears just on the horizon, with devices like the Meta Quest Pro and the Apple Vision Pro available to consumers. These devices all have cameras pointing out into the world, but as of now, they're just used for features such as hand and environment tracking. We'd like to see if these cameras can act as a "second pair of eyes", using new developments in LLMs and Vision models to help you process and learn about the world around you. To achieve this goal, we plan on building a model that, given an egocentric video, can answer questions about the scene that it sees.

More explicitly, we wish to build a model such that given a question and a video, it can output the correct answer to the question given a set of answer choices.

2. Problem Statement

At its core, our problem can be framed as a Video Question-Answering problem. We are seeking to take as input the video feed of what the user is seeing, as well as questions that the user is posing, and return a valid answer to that question. Framed a bit more precisely, the goal of the Video QA algorithm is to predict an answer a* given a video V and a question Q. When predicting a*, two approaches that have been taken is to give the model multiple choices to choose from, such that $a^* \in A$ (where A is a set of potential answers), or give the model no choice at all, computing correctness based on the model's output answer a' to the correct answer a*.

2.1. Background

Video QA is a well-explored topic in the Computer Vision field, from Xu, et al.[7], a large-scale video benchmark, created for the purpose of translating videos to text, to many more since then. Lots of approaches have been taken to solve this problem as well, from spatiotemporal attention in Jang, et. al [12] to hierarchical graphical models, as proposed by Cherian, et. al [1]. What makes this task difficult, beyond the challenges of image classification and segmentation, is that answers to the posed questions often require an understanding of not only the objects and activities present in the scene, but also the underlying spatial, temporal, and causal relationships that underlay the scene.

Another important aspect of Video QA is the incorporation of external knowledge sources. For instance, Gupta et al. proposed an approach called "Knowledge-Aware Video Question Answering"[10], which leverages external knowledge graphs to enhance the reasoning capabilities of the model. By incorporating structured knowledge, such as facts about objects, actions, and relationships, these models can effectively reason about complex questions requiring deeper understanding.

Furthermore, attention mechanisms have proven to be valuable in Video QA. Nam et al. introduced the "Dual Attention Networks"[5] that jointly model spatial and temporal attention to highlight informative regions and video frames for answering questions. By dynamically focusing

on relevant visual and temporal cues, these models can better understand the context and provide more accurate answers.

2.2. Socratic Models

Traditionally, Visual QA problems have been addressed by training end-to-end models that take in the image and the query jointly, and predict the correct answer to the query at the end. These models usually require large amounts of training data, and still often perform poorly when asked questions outside their distribution. Foundation models, however, solve some of these problems. They are trained on large swaths of data over many domains, allowing them to adapt well to many downstream tasks. However, among these many foundations models, the domains of their data rarely overlap. Large language models, for example, are often trained on text from the internet and are not accompanied by images, as seen in Figure 1. Visual language models, on the other hand, are trained on internet-scale image-caption pairs but are often missing the context that may surround the image.

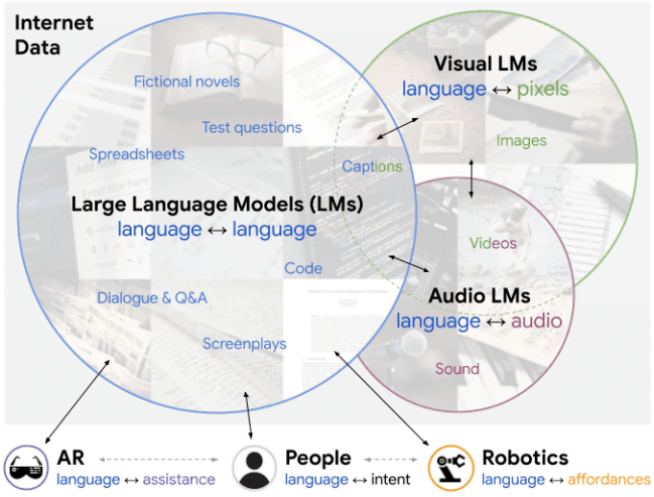

Figure 1. Socratic Models - An example of how foundation models trained across different domains often have little overlap in their domains

The Socratic Models paper[6], presents a way to chain these models together, such that the inputs of one can be fed into another. More precisely, multiple large pre-trained models are composed through language (via prompting) without requiring training, to perform new downstream multimodal tasks. The composed models communicate with each other purely through natural language (using machine learning terms, natural language acts like the latent representation between the different models). In the Socratic Models paper, they refer to this representation as the World State. With this in mind, we can frame Video Question Answer as a downstream task that relies on multimodal information - the understanding of a video sequence, and the ability to bring in external context (including egocentric context) to answer a given query about the image. This all culminates in an approach that reaches state-of-the-art results, all while leveraging existing foundation models with zero-shot capability.

2.3. OFA Foundation Model

In order to use the method presented in the Socratic Models paper, we must use capable foundation modules that interact with one another. The OFA (One-For-All) model[9] is a multimodal foundation model that achieves SOTA results on a wide variety of multimodal tasks, including image captioning, which would be applicable to our task. OFA is built as an encoder-decoder architecture with transformer blocks and was trained on a wide variety of multimodal tasks, as shown in 2.

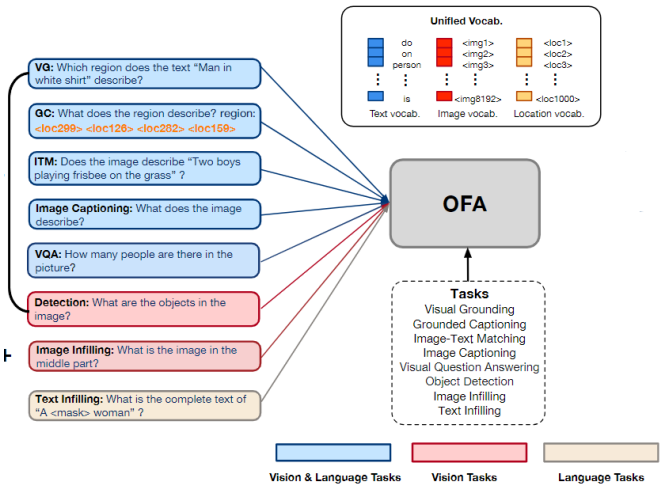

Figure 2. OFA Multimodal multitask training

2.4. Egocentric Data

What makes our problem somewhat unique is the input of egocentric data. Traditionally, Video QA is trained and tested on 3rd person videos, such as youtube videos or videos taken by an observer looking at a scene. However, since our goal is to build an algorithm that would hopefully be deployed on some type of Augmented Reality Device, the input video is from the user's perspective. This is often called ego-centric video. In most cases, this difference in perspective doesn't create a meaningful difference in the scene. For example, if the user is looking at a cat jumping onto a couch, that view will be the same for ego-centric and non-ego-centric data. The difference comes in when the user themselves is interacting with an object or doing a task. The act of cutting onions is much different from a user's perspective vs. an outside viewer's perspective.

What makes this task even more difficult is the lack of data available. 3rd person footage is now relatively easy to come across through platforms such as youtube, but egocentric video is much rarer and often needs to be deliberately collected. However, certain datasets do exist, and we'll discuss which one we chose and why in the next section.

2.5. Dataset

While egocentric data is sparse compared to video data as a whole, a few high-quality datasets do exist. For our project, we chose the EgoVQA [4], a fully egocentric video dataset with 500+ videos, each with a question and 5 potential answer choices. The egocentric videos are each separate, and thus each has its own context associated. Many of the videos and questions rely on some of the unique aspects of egocentric data, such as asking about the user themselves. For example, one question in the dataset asks "How many people am I talking with", which requires not only an understanding of the number of people in the image but also an understanding of who the "I" in the image might be.

For the purposes of our project, we followed the train/val/test split recommended in the paper, with 250 videos used for the training set, 150 for the validation, and 120 for the final test.

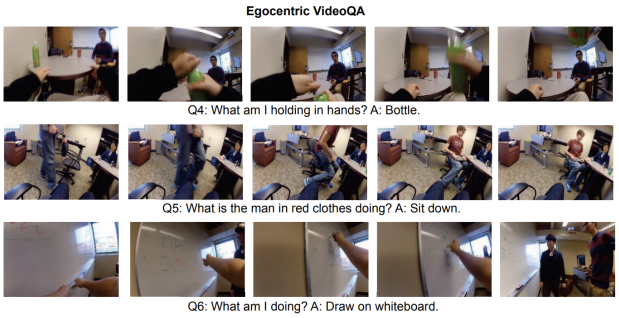

Figure 3. Collection of Egocentric pictures from EgoVQA

.

3. Methods

3.1. Modules

In order to achieve the task of Video Question-Answering, we separated our system into a vision module, a world state, and an answering module:

1. **Vision Module**: A vision language model that takes video as input, and outputs a world state that describes key features and relationships of the video.

2. **World State**: Text from the Vision Module that is formatted to capture essential information of the video as it relates to the question. This format is inspired by the Socratic Models paper, to make the World State a latent representation used between the two modules.

3. **Answering Module**: A language model that takes in both the question and the World State, and outputs an answer to the question.

3.2. Baseline Method

Our baseline approach aims to use off-the-shelf models that are not trained for our specific task, to see how well existing models perform on video question answering.

For the baseline Vision Module, we used a general-purpose image captioning model built on HuggingFace's Vision Encoder Decoder structure. The encoder is a Vision Transformer model [3], the decoder is GPT-3, and the model was trained on the COCO Dataset [11]Ḃecause this model is for just image captioning (not video), we selected a frame in the middle of the video for the model to run inference on.

For the baseline Answering Module, we used OpenAI's Text-Davinci-003 model.

3.3. OFA Model - PromptCap

The main disadvantage of the baseline method is that the caption (which we can call the World State) generated by the Vision Module does not take into account the question, and so it might oftentimes leave out information in its caption that is vital to answer the question correctly. We decided to try a new model for the vision module, the OFA (One-For-All) multimodal model presented in 2.3. We used weights from a finetuned version of OFA specifically meant for caption generation based on a query, which was presented in the paper PromptCap [13]. PromptCap takes as input an image and a query and then outputs a caption that would help answer the query.

3.4. Finetuning PromptCap

However, PromptCap is trained entirely on 3rd person-view images. This allows it to perform well on pictures taken from a camera or from an external perspective but doesn't address some of the key limitations of egocentric data, as discussed earlier. For this reason, we will finetune PromptCap using a subset of our training data, withholding the rest of our training data for the other methods described below. However, our training data only contains questions, answers, and videos. To train PromptCap to generate good captions, we ideally would have egocentric video-caption pairs. But in the absence of this data, we took a similar approach that PromptCap itself used to generate its own synthetic image-caption pairs.

First, we used a purpose-built caption generator, BLIP[8], to generate captions for 3 frames in each video sequence. Then, we took the question-answer pair for each

frame (corresponding to the video sequence it was drawn from) and concatenated it with the BLIP caption. This resulted in a list of tuples of the form (frame, (BLIP caption, question, answer)). We then fed these frame by frame into GPT3 using a specially crafted prompt with examples to follow. This resulted in summaries for each frame, which took into account both the BLIP caption and the largely egocentric questions from our dataset (questions such as, what am "I" holding, what am "I" doing, etc.). This process is shown in Figure 4

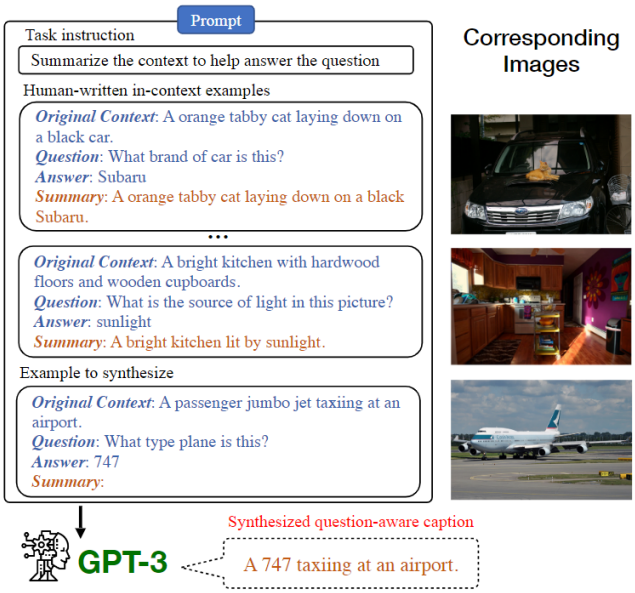

Figure 4. Synthetic caption generation via GPT Prompting

Now that the dataset for finetuning had been prepared, next came the finetuning itself. As mentioned earlier, PromptCap follows a traditional Encoder-Decoder Transformer architecture. As the model has been massively pretrained, and our data points were relatively minimal in comparison, we avoided finetuning the model in its entirety. Instead, we opted to freeze the majority of the model and only modify a small subsection of the model. As the main difference distribution-wise between the input data and our egocentric data is the presence of "I" variables and the presence of first-person hands and such, we opted to unfreeze that last layer of the encoder (which includes a Transformer as well as a fully-connected net). The idea here was that unfreezing the transformer would allow the attention weights to update, and attend to words that may be more important in egocentric captioning as compared to normal captioning.

3.5. PromptCap Variations

Normally, PromptCap takes into account a query that can aid the generation of captions. We extended its use case by testing different combinations of information given to the model. We tested out prompts that the authors of PromptCap suggested, and also tried different variations of supplying the question and answer choices to the models. These approaches are described in the following sections, and summarized in the table 1.

Information given	*Query to PromptCap*
Prompt	What does the image describe?
Prompt + Question	Describe the image in a way that would help ChatGPT answer the question: [question]?
Prompt + Question + Choices	Describe the image in a way that would help ChatGPT answer the question: [question]? Where the answer choices for the question are: [choices]
Question	[question]?
Question + Choices	[question]? Your choices are: [choices]

Table 1. PromptCap Query Variations

3.5.1 PromptCap to Generate Captions

Our first set of methods in using PromptCap includes using PromptCap in order to generate a caption (i.e. generate a world state) that our LLM can use in order to answer the Visual QA. In this set, there are two variations.

The first variation is where only a prompt is fed into PromptCap, i.e. "What does the image describe?" Based on the way PromptCap was trained and finetuned, this will lead PromptCap to generate a generic caption for the provided image.

The second variation goes further and prompts PromptCap to generate a caption tailored to the question that needs to be answered. This is framed using the following template: "Describe the image in a way that would help ChatGPT answer the question: [question]?" Ideally, this would lead to a caption that contains the answer to the question.

In both of these variations, the generated caption, along with the question, is then sent to the LLM, which is prompted to answer the question. To help it with this task, the LLM is provided 5 answer choices to choose from, of which only one is the correct answer.

3.5.2 PromptCap to Answer the Question Directly

Another set of methods is to provide PromptCap with the question directly and allow it to answer the question provided, cutting out the middle part of providing a caption to GPT. The output of PromptCap is still provided to GPT, where it takes the answer provided by PromptCap into account as it is choosing its answer. Due to our finetuning in addition to the QA pretraining of PromptCap itself, this

may perform better than the previous method of PromptCap generating a caption, as the necessary the model can simply refer back to the image (as it is using a transformer mechanism), as opposed to creating a static caption that may not capture all the necessary information.

Within this set of methods, we have two variations. The first variation is to present PromptCap with the question directly. Its generated caption/answer is then provided to GPT, along with the question again, as well as the answer choices. The second variation is to provide PromptCap the questions along with the answer choices, and again provide its caption/answer to GPT along with the question and answer choices. We expect the version with the answer choices to perform better but are curious to what degree it performs better.

3.6. Augmenting the World State

For all of our experiments thus far, we have built our world state off of just one frame in the video. In this section, we extend upon an idea presented in the Socratic Models paper, where they construct the world state off of multiple frames with timestamps, to encode the World State with more detail and information about motion throughout the video, as in the example below:

> Timestamp 1: Two people sitting down and talking
> Timestamp 2: One person in a blue shirt and another in a red shirt
> Timestamp 3: An open door and one person sitting down

With this method, we tested out different numbers of timestamps used in the world state to see its impact.

4. Experiments

For all experiments, we ran the models on the validation dataset of EgoVQA and measured the accuracy of answering the questions correctly. We chose accuracy as our main evaluation metric, evaluated as the number of correct answers over the number of total answers. We chose this as it's a straightforward metric that captures the overall efficacy of the model, and it is the metric used in most VQA literature.

Additionally, we evaluated the section below on the held-out test portion of our dataset that hadn't been touched prior to the final evaluation. We did this to make sure that our models weren't preconditioned to guess the right answer based on the training data. We also sampled the test set randomly, such that videos from across the dataset were included (if we had just picked the last x number of videos, those videos would have been highly correlated due to the structure of the EgoVQA dataset).

4.1. Evaluating Vision Modules

First, we evaluated different iterations of the vision module, including the baseline ViT model and variations of the promptcap model. The first model shown in the table is the baseline, and the second model is the out-of-the-box version of PromptCap. The rest of the models are variations of the PromptCap model after it was finetuned on the EgoCentric Video dataset. The variations are outlined in 1. Finally, the last model in the table is the original SOTA benchmark when the EgoVQA dataset was released, from the ST-VQA model discussed in the background section.
The results from the experiments can be seen in Table 2.

Model	Accuracy
Baseline	12%
Prompt (Unfinetuned)	17%
Prompt	22%
Prompt + Question	29%
Prompt + Question + Choices	30%
Question	38%
Question + Choices	**52%**
ST-VQA	38%

Table 2. Experiments

From the results, we immediately see that the baseline has very poor performance, getting 11% accuracy. In the dataset itself, there are only 5 answer choices per question, so a model that answers at random should get at least 20% accuracy. We realized that because the World State for the baseline model was poor, the answering module would often output phrases like "I'm sorry, as an AI language model, I cannot see the scene you are currently looking at and cannot determine who is entering the room through the door" instead of guessing on the question. We tried to enforce this with better prompt engineering for the Answering Module but with little success. We realized much of this was due to unfamiliarity with egocentric data, and thus our next step was to finetune.

After the finetuning process, as described in the section above, the accuracy increased by 5% as compared to the model that hadn't been finetuned. As the only thing that changed between the two experiments was the fine-tuning, and due to our separation of the validation and the training sets, we can attribute this increase in performance to the finetuning itself. Due to this increase in performance, for all experiments going forward (i.e the various manipulations of the prompt and the world space), the OFA model used was the OFA model finetuned with PromptCap and further finetuned by us using Egocentric Video data.

We see that all variations of the PromptCap model performed better than the baseline and better than random guessing (20%). Additionally, it seems that the prompt,

something like "Describe the image in a way that would help ChatGPT answer the question.." hindered the model, and instead simply asking the question itself yielded better results at 38% accuracy. Finally, giving the PromptCap model the questions and the answers but no additional prompt yielded the best results of 52%. As an interesting comparison, our best-performing model achieves significantly higher results on the task compared to the original SOTA model from the dataset paper, ST-VQA.

The results show that giving the PromptCap model the most amount of relevant information about the question and the answer choices yields the best results. Additionally, it seems that additional semantics in the prompt actually decrease the overall accuracy. This also aligns with the way the PromptCap model finetuned OFA, in which it was given just the barebones question. It would be interesting to construct alternative ways to finetune OFA for our specific task that allows for more freeform prompt engineering to guide the response.

When evaluating different iterations of the vision module, we noticed that the world states generated were rather concise, only containing what it deemed to be relevant information. This might have been another limitation of the way in which PromptCap finetuned the OFA model. For further testing, we could consider doing separate initial finetuning of the OFA model that encourages more verbose captions.

4.2. Evaluating World State Augmentation

Next, we ran experiments on augmenting the World State to capture multiple frames in the video, as discussed in 3.6. We used the best-performing vision module from the previous section, which was the Question + Choices version of PromptCap, and then varied the number of frames used in the world state, where frames were selected to be equidistant from one another in the video. The results can be shown in Figure 5

From the results, we see that augmenting the world state with multiple frames in fact slightly decreased the performance of the best model. Upon inspecting the world state and different correct and incorrect responses, it seemed that the augmentation did yield better results for some videos in which a lot of different activities were happening, but oftentimes also did worse by seemingly inundating the Answering Module with too much information and confusing it. Although the results of this approach did not seem promising in our experiments, we would like to explore the general idea of augmenting the world state further, as it was shown to be successful in the Socratic Models paper.

5. Conclusion

In conclusion, in this paper, we were able to approach the problem of Video QA on Egocentric data using the relatively new framework established in the Socratic Models paper. More specifically, we composed a Vision Language Model and a Large Language Model to answer questions about a user's environment from a first-person point of view. Using the OFA model, finetuned via the PromptCap paper, and further finetuned by us using Egocentric Videos with custom generated captions, as well as GPT-3, we were able to create a framework that would pick the correct answer roughly 52% of the time, if it was given both the question and the answer choices. This is significantly above the expected probability if the model were picking at random (which is 20%), More excitingly, however, we far outperform the state of the art (at that time) presented in the original EgoVQA paper[4], which has an overall accuracy of 37.57%. Additionally, it also outperforms a newer paper that attempts to boost accuracy on the EgoVQA dataset through a myriad of data augmentation techniques[2]. This paper by Falcon et. al. posts an overall accuracy of 37.71%, and a category best of 47.62%, which is still worse than our best-performing model.

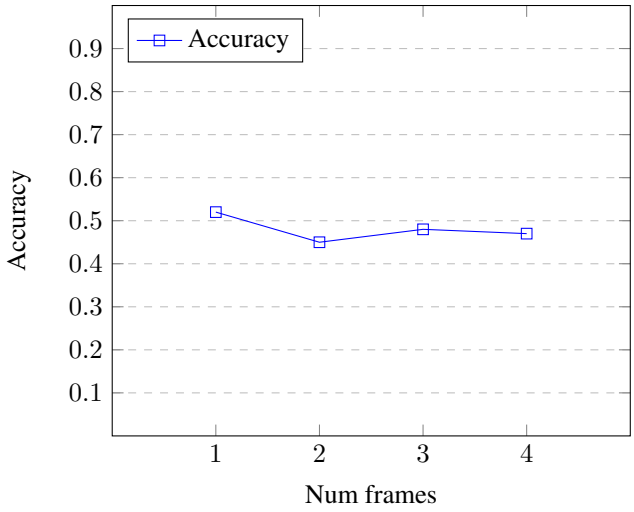

Figure 5. World State Augmentation

5.1. Next Steps

There are several next steps we could take to improve this model. First and foremost, we mainly would focus on temporal reasoning of the video data. For the most part, our models looked at a single frame in the video in order to reason and answer the given question. Even when we explored giving the model multiple frames, the single-frame version performed better. However, the model was not able to capture motions and actions, such as "sitting down" or "using a phone."

Additionally, we wish to extend this framework to a wider use case for Video Question Answering. Our initial motivation for the project was to build a general reasoning

platform that can be used with AR/VR headsets as a "second pair of eyes". As such, we plan build a demo on AR hardware to showcase the technology.

6. Contributions

Arjun and Ronak both contributed equally to the project, working together on the data preprocessing, finetuning of the models, all other engineering tasks, and writing the paper.

7. GitHub

Check out our GitHub https://github.com/rmalde/Ego-QA-231 to try out the baseline implementation and reproduce the results.

References

[1] T. M. A. Cherian, C. Hori and J. L. Roux. (2.5+ 1) d spatio-temporal scene graphs for video question answering, 2022.

[2] G. S. Alex Falcon, Oswald Lanz. Data augmentation techniques for the video question answering task, 2020.

[3] A. K. D. W. X. Z. T. U. M. D. M. M. G. H. S. G. J. U. N. H. Alexey Dosovitskiy, Lucas Beyer. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[4] C. Fan. Egovqa - an egocentric video question answering benchmark dataset, 2019.

[5] J. K. Hyeonseob Nam, Jung-Woo Ha. Dual attention networks for multimodal reasoning and matching, 2017.

[6] J. K. Hyeonseob Nam, Jung-Woo Ha. Socratic models: Composing zero-shot multimodal reasoning with language, 2022.

[7] T. Y. J. Xu, T. Mei and Y. Rui. Msrvtt: A large video description dataset for bridging video and language, 2016.

[8] D. R. Junnan Li, Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022.

[9] R. M. J. L. S. B. Z. L. J. M. C. Z. J. Z. H. Y. Peng Wang, An Yang. Ofa: Unifying architectures, tasks, and modalities, through a simple sequence-to-sequence learning framework, 2022.

[10] M. G. Pranay Gupta. Newskvqa: Knowledge-aware news video question answering, 2022.

[11] A. Singh. The illustrated image captioning using transformers, 2022.

[12] Y. Y. Y. K. Y. Jang, Y. Song and G. Kim. Tgif-qa: Toward spatiotemporal reasoning in visual question answering, 2017.

[13] Z. Y. W. S. N. A. S. J. L. Yushi Hu, Hang Hua. Promptcap: Prompt-guided task-aware image captioning, 2022.