Visual Question Answering

By

Osama Khaled Abdelghaffar	20190086
Paula Adel Kamal	20190139
Mahmoud Amr Mahmoud	20190499
Nedal Adham Ahmed Ezz-Eldin	20190585
Youssef Osama Ahmed Moemen	20190629
Mohammad Alameen Abdilaziz	20190720

Supervised by
Dr. Motaz Elsabban

Artificial Intelligence Department
Cairo University

2022-2023

Acknowledgments

We would like to express our deep and sincere gratitude to our research supervisor, Prof. Motaz Ahmed Zaki Elsabban, for giving us the opportunity to conduct research and providing invaluable guidance throughout this work. His dynamism, vision, sincerity, and motivation have deeply inspired us. He has taught us the methodology to carry out the work and to present the works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

We are greatly indebted to our honorable teachers of the Department of Artificial Intelligence at Cairo University who taught us during the course of our study. Without any doubt, their teaching and guidance have completely transformed us into the persons that we are today.

We would like to extend our heartfelt gratitude to Huawei for their generous support in providing us with free computing resources throughout our research project. The computing resources offered by Huawei played a crucial role in enabling us to carry out our experiments and analyze the data effectively. We sincerely appreciate their commitment to fostering technological advancements and their dedication to supporting academic research.

We would like to express our sincere gratitude to Salesforce for making their model BLIP public, which proved to be an invaluable resource throughout our research. The availability of BLIP significantly enhanced the quality and efficiency of our work, enabling us to leverage its capabilities in various aspects of our project. We extend our heartfelt appreciation to Salesforce for its commitment to advancing the field of artificial intelligence and for providing researchers like us with access to cutting-edge tools and technologies.

Additionally, we would like to extend our thanks to Shuang Li, the lead author of the paper "Composing Ensembles of Pre-trained Models via Iterative Consensus." Their work served as a guiding light for us during the implementation of the PIC (Pre-trained Iterative Consensus) model. Shuang Li's valuable insights and recommendations helped us navigate the intricacies of the PIC model, leading us in the right direction and enabling us to achieve significant progress in our research. We are truly grateful for their expertise and support.

We are extremely thankful to our parents for their unconditional love, endless prayers and caring, and immense sacrifices for educating and preparing us for our future. We would like to say thanks to our friends and relatives for their kind support and care.

Finally, we would like to thank all the people who have supported us to complete the project work directly or indirectly.

Date: June 08, 2023

Abstract

This document delves into the realm of Video Question Answering (VideoQA) and Image Question Answering (IQA), providing a comprehensive view of datasets, problem formulation, and state-of-the-art models used in these domains. For VideoQA, we utilize a PIC framework that leverages a FrozenBiLM model for zero-shot VideoQA. However, the model's performance is influenced by video content and caption quality. To enhance alignment between the captions and questions, we proposed an approach using the user's question as a caption prefix for each video frame.

For IQA, we focus on the VQA2 and VizWiz datasets, particularly the latter, which is designed for visually impaired individuals. Using a BLIP architecture, we achieved an accuracy of 87.30% on the VizWiz validation set, significantly exceeding the baseline model's accuracy. However, the model demonstrated limited generalization capabilities on images sourced from the internet, underscoring the need for diversity in training data.

Despite promising results, significant challenges were encountered, particularly the need for substantial computational power and the unique challenges presented by the VizWiz dataset. Future work involves harnessing additional computational power, incorporating diverse modalities, and fine-tuning models for specific domains to improve both accuracy and efficiency in VideoQA and IQA tasks.

Our work also experiment in the VideoQA partition which were we implemented a composing ensembles of pretrained models using iterative consensus the framework manily using multiple clip models with GPT2 to caption a video and then pass the caption along the question to the openai GPT3.5-turbo to answer the question.

Contents

List of Tables

List of Figures

Abbreviations list

- **VQA** - Visual Question Answering
- **BLIP** - Bi-directional Language Image Pre-training
- **VQA-Med** - Visual Question Answering in the Medical Domain
- **PIC** - Composing Ensembles of a **P**re-trained Models via a **I**terative a **C**onsensus
- **MM-VQA** - Multi-modal Visual Question Answering
- **KB-VQA** - Knowledge-based Visual Question Answering
- **MC-VQA** - Multi-choice Visual Question Answering
- **OE-VQA** - Open-ended Visual Question Answering
- **WUPS** - Wu-Palmer Similarity
- **BiLM** - Bidirectional Language Model
- **COCO** - Common Objects in Context
- **CLIP** - Contrastive Language-Image Pretraining
- **OCR** - Optical Character Recognition
- **LM**: Language Model
- **GPT**: Generative Pre-trained Transformer
- **CE**: Cross Entropy
- **ITC**: Image-Text Contrastive Loss
- **ITM**: Image-Text Matching Loss
- **LM**: Language Modeling Loss
- **BLIP**: Bidirectional Language Image Processing
- **MED**: Multimodal Mixture of the Encoder-Decoder
- **MLP**: Multilayer Perceptron
- **BERT**: Bidirectional Encoder Representations from Transformers
- **SA**: Self-Attention
- **FFN**: Feed-Forward Network
- **CA**: Cross-Attention
- **ACC**: Accuracy
- **GPU**: Graphics Processing Unit
- **VRAM**: Video Random Access Memory
- **RAM**: Random Access Memory
- **AI**: Artificial Intelligence
- **NLP**: Natural Language Processing
- **LLMs**: Large Language Models
- **VALOR**: VVision-Audio-Language Omni-Perception
- **VAST**: Video and Audio Scene-aware Transformer
- **WildQA**: Wildly Multimodal Video Question Answering

1 Introduction

Visual Question Answering (VQA) serves as a crucial research field that combines the principles of computer vision and natural language processing. The goal is to create sophisticated algorithms capable of responding to questions phrased in natural language, which pertain to visual content such as images or videos. The complexity of this task lies in the necessity for machines to comprehend both the visual elements and the accompanying descriptive language.

The task encompasses providing answers to open-ended or multi-answers queries rooted in a video or image. These responses are also expected to be in natural language, thereby mimicking the human ability to provide natural responses to questions. The utility of such a system is evident in its ability to answer questions related to any given visual content, be it an image, video, or infographic.

Building on the concept of Image Question Answering, Video Question Answering introduces a layer of complexity by including videos. This includes the identification of objects, actions, and the relationships between different objects, as well as more abstract ideas such as emotional states or sentiments.

Image Question Answering can serve as a powerful tool in educational environments, aiding students in their learning journey by answering questions about visual content. Additionally, in a healthcare setting, it can support doctors in their diagnostic processes through the analysis of medical images.

Video Question Answering pushes the envelope further by analyzing video content and generating relevant responses. The applications of this technology are vast and include video search, video captioning, and video retrieval. This capability can revolutionize various fields, including entertainment, education, healthcare, and security. Video Question Answering (VQA) and Image Question Answering systems hold considerable potential for a broad range of applications. For instance, in the context of a video, these systems can assist in tasks such as video indexing, video retrieval, video summarisation, managing learning systems, and analyzing surveillance footage. These are merely a few examples of the myriad potential uses for video question-answering technology.

Likewise, Image Question Answering systems can significantly contribute to various scenarios. In the education sector, these systems could enhance experiences at venues like museums by empowering visitors to pose direct questions about exhibits, akin to interacting with ChatGPT. Moreover, these systems can prove invaluable in refining Image Retrieval operations. For example, if a user queries, "do you observe a door," the system could pull up all images containing a door from a given dataset.

Despite the tremendous potential and practicality of VQA systems, they are not without their challenges. Key obstacles in the development and implementation of VQA systems include accommodating language and visual content variations, managing ambiguities in questions and answers, and designing models that can adapt and generalize across new domains. Visual Question Answering (VQA) models also need to navigate the challenges posed by fluctuations in lighting, shifts in perspective, and other visual factors. These factors could interfere with the accuracy of the models, especially when dealing with blurred images or video frames. Moreover, VQA models typically require an extensive amount of labelled data for effective training, and acquiring such data can be both challenging and costly. Consequently, the absence of large-scale datasets for training and testing these models is one of the most significant obstacles.

In the realm of Image Question Answering, several models have been developed, including the BLIP model. We have fine-tuned this model using the Vizwiz dataset to enhance its performance.

Introduced in 2021, the BLIP (Bi-directional Language Image Pre-training) model employs a deep learning approach, specifically designed for the task of VQA. The model, based on the transformer architecture, is adept at learning representations from both language and images, enabling it to excel in VQA tasks. The model's pre-training on a substantial dataset of image-caption pairs equips it to map images to natural language descriptions and vice versa.

Much like other transformer-based models such as BERT and GPT, the BLIP model uses self-attention to process input sequences. However, a distinctive feature of the BLIP model is its inclusion of a visual encoder that processes image features and integrates them into the model's representations. One of the key strengths of the BLIP model lies in its capability to manage lengthy and intricate questions. The BLIP model's design caters to the processing of questions that contain up to 40 words. This feature allows it to manage intricate queries that necessitate multiple steps to resolve. Furthermore, the model incorporates a multi-hop attention mechanism. This facilitates the model's ability to reason across multiple data points within the image and language input.

The BLIP model has earned recognition for delivering state-of-the-art results across several benchmark VQA datasets, such as VQA 2.0, GQA, and CLEVR. Demonstrating its versatile applicability, the model has also shown proficiency in adapting to new domains. An instance of this adaptability is its performance within the field of medical imaging, where it has achieved remarkable results on the VQA-Med dataset.

As a tool for VQA tasks, the BLIP model has catalyzed significant advancements in the realms of computer vision and natural language processing. The model's competence in managing complex, lengthy questions and its ability to reason with multiple pieces of information position it as a promising resource for diverse applications. These range from education and healthcare to entertainment and social media. These remarkable attributes have prompted us to opt for the BLIP model, anticipating its potency to enable us to deliver exceptional performance in Image Question Answering.

The VIZWIZ dataset was established to promote research in Visual Question Answering (VQA) specifically for individuals with visual impairments. This dataset is comprised of real-world images taken by visually impaired individuals, alongside associated questions and answers. It was primarily devised to address the scarcity of data suitable for training and assessing VQA models catering to people with visual impairments. The data was compiled through a mobile app that empowered visually impaired individuals to photograph objects or scenes and pose questions about them. These inquiries could pertain to the identification of objects within the image or their spatial relationships. The answers were subsequently provided by a crowd-sourced assembly of human annotators.

For our Image Question Answering project, we chose to concentrate on the VIZWIZ dataset and fine-tune the BLIP model accordingly. Such a decision is expected to enhance the BLIP model's performance in a manner that aligns with our methodological approach.

Contrastingly, for Video Question Answering, there are relatively fewer models available, and their performance has generally been subpar. As a result, we decided to develop our model, implementing the PIC (Proposal-Identification-Verification) model. This deep learning model is designed to tackle VQA challenges such as handling variations in language and visual content, and dealing with ambiguity in questions and answers.

Our findings indicate that the PIC model is an exceptional asset for Video Question Answering, having achieved state-of-the-art results across multiple benchmark datasets. However, the implementation of this model presents certain complexities. It necessitates substantial amounts of training data and specialized hardware for the effective training and execution of deep neural networks.

2 Literature Review

We will divide this section and pretty much all the upcoming sections into two subsections, one focusing on the Video Question Answering and the other Focusing on the Image Question Answering

2.1 Video Question Answering

2.1.1 Datasets

[Zhong et al., 2022] categorize VideoQA datasets into three taxonomies related to the modality, Question type, and video type. Normal VideoQA, multi-modal VideoQA, and knowledge-based VideoQA are the types based on the modalities invoked in the question-answer pairs, normal VQA is concerned only with the video frames while MM-VQA can take another modality as subtitles or movie plots as [Tapaswi et al., 2016]. As for KB-VQA, it uses a knowledge base to ask general questions that may not be in the video itself but rely on common knowledge as in [Garcia et al., 2020]

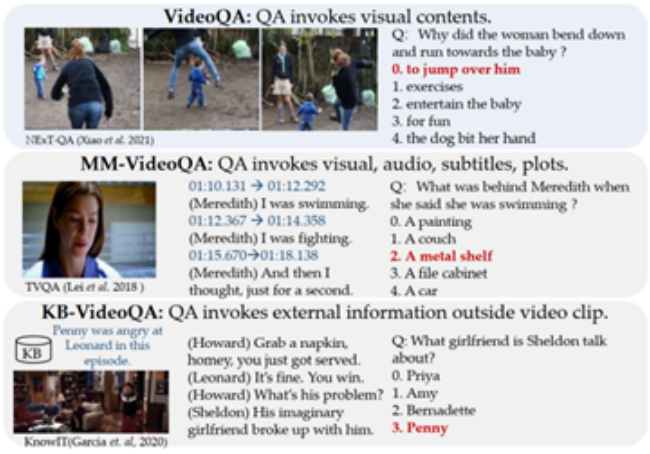


Figure 1: Illustration of VideoQA, Multimodal VideoQA (MM) and Knowledge-Based VideoQA (KB-VQA)

Factoid VideoQA and inference VideoQA are the types based on the type of question the dataset is concerned with, Factoid VQA is concerned with a fact question like what, who, how many, etc. while Inference VQA is concerned more with causal and temporal questions about the video as question words like how, why and when. Most of datasets deal with Factoid VQA but more recent work began working on inference as [Yi et al., 2020; Xiao et al., 2021].

Multi-choice QA and open-ended QA are the types based on the answers existing in the datasets. MC-VQA contains with every question answer pair other wrong choices and the model

is trained to select the correct answer, while OE-VQA contain pairs of open ended questions and answers.

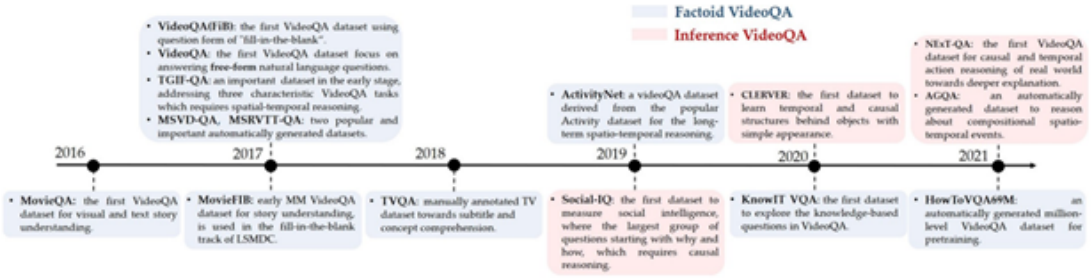


Figure 2: Historical evolution of datasets. Datasets of VideoQA are listed above the timeline. Datasets of Multi-modal and Knowledge-based VideoQA are listed below the timeline. Blue and red color represents datasets focused on Factoid VideoQA and Inference VideoQA.

They also look at the dataset according to the way they were annotated either If it was annotated completely by humans or scraped from the web with the percentage of human annotation or completely scraped from the web.

Dataset	MB	QB	Data Source	Goal	#Video	#QA	Annotation	QA Task
VideoQA(FiB) [Zhu *et al.*, 2017]	Vid	F	Multiple source	Temporal reasoning	109K	390K	Auto	MC
VideoQA [Zeng *et al.*, 2017]	Vid	F	Web videos	Description	18K	174K	Auto, Man	OE
TGIF-QA [Jang *et al.*, 2017]	Vid	F	GIF	Spatio-temporal reasoning	71K	165K	Auto, Man	MC, OE
MSVD-QA [Xu *et al.*, 2017]	Vid	F	Web videos	Description	1.9K	50K	Auto	OE
MSRVTT-QA [Xu *et al.*, 2017]	Vid	F	Web videos	Description	10K	243K	Auto	OE
ActivityNet-QA [Yu *et al.*, 2019]	Vid	F	Web videos	Description	5.8K	58K	Man	OE
MovieQA [Tapaswi *et al.*, 2016]	MM	F	Movies	Text & visual story comprehension	6.7K	6.4K	Man	MC
MovieFIB [Maharaj *et al.*, 2017]	MM	F	Movies	Description	118K	348K	Auto	OE
TVQA [Lei *et al.*, 2018]	MM	F	TV shows	Subtitle & concept comprehension	21K	152K	Man	MC
HowToVQA69M [Yang *et al.*, 2021]	MM	F	Web videos	Pre-training for downstream tasks	69M	69M	Auto	OE
KnowIT VQA [Garcia *et al.*, 2020]	KB	F	TV shows	Knowledge in VideoQA	12K	24K	Man	MC
Social-IQ [Zadeh *et al.*, 2019]	MM	I	Web videos	Measuring social intelligence	1.2K	7.5K	Man	MC
CLEVRER [Yi *et al.*, 2020]	Vid	I	Synthetic videos	Temporal and causal structures	10K	305K	Auto	MC, OE
AGQA [Grunde-McLaughlin *et al.*, 2021]	Vid	I	Homemade videos	Compositional reasoning	9.6K	192M	Auto	OE
NExT-QA [Xiao *et al.*, 2021]	Vid	I	Web videos	Causal & temporal action interactions	5.4K	52K	Man	MC, OE

Figure 3: Table made by [Zhong et al., 2022] containing Video QA datasets (MB: Modality-based Taxonomy, QB: Question-based Taxonomy, Vid: VideoQA, MM: Multi-modal VideoQA, KB: Knowledge-based VideoQA, F: Factoid VideoQA, I: Inference VideoQA, Auto: automatic annotation, Man: manually annotation, MC: multi-choice QA, OE: open-ended QA.)

The paper also discusses the evaluation metrics used in VideoQA, such as accuracy and WUPS (Wu-Palmer Similarity is a soft measure of accuracy that takes into account word synonyms). It is calculated by using this equation:

$$WUPS = \frac{2 \times Depth(LCS)}{Depth(word1) + Depth(word2)} \tag{1}$$

In this equation:

- Depth of LCS or Lowest Common Subsumer refers to the depth (distance from the root) of the closest shared ancestor in the hierarchy.
- Depth of word 1 and Depth of word 2 are the depths of the respective concepts in the hierarchy.

The Wu-Palmer similarity ranges from 0 to 1, where 0 indicates no similarity, and 1 represents the highest similarity between the concepts or words.

2.1.2 Techniques and Methods

Normally we deal with VQA with four components, video encoder, question encoder, cross-modal interaction, and answer decoder.

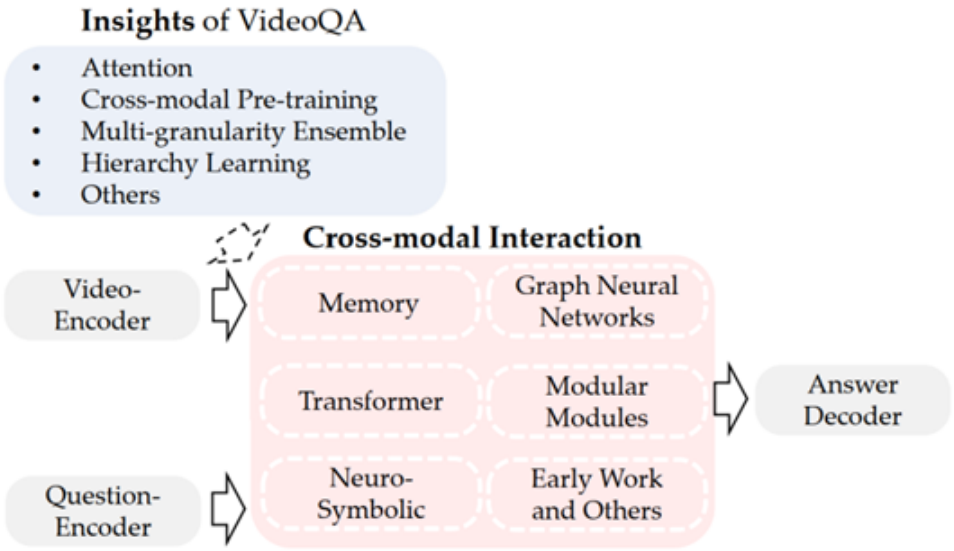


Figure 4: A common framework of VideoQA, consisting of four components: video encoder, question encoder, cross-modal interaction, and answer decoder

We have read multiple research papers on models used in VQA that use this kind of architecture most ones that have promising results pre our work are

1-FrozenBiLM or Frozen Bidirectional language Model yang2022 is a framework for multi-modal inputs that extends frozen bidirectional language models. It is designed to tackle zero-shot VideoQA through masked language modeling.

The model consists of a large frozen bidirectional language model (BiLM) and a frozen pre-trained visual encoder, complemented with additional lightweight trainable modules. These modules include a visual-to-text projection module P, which maps the frozen visual features to the joint visual-text embedding space, and a set of small adapter modules A in between the frozen transformer blocks. The pre-trained normalization layers in the BiLM are also finetuned. The model is trained using Web-scraped data and Masked Language Modeling. Here are the steps of the algorithm in more detail:

1. **Language Model Pre-training:** The method employs a pre-trained language model, which is tokenized and converted into a continuous D-dimensional embedding. This language model is pre-trained on a vast text dataset from the web. During the training process, the pre-trained language model remains static, which is found to be essential for zero-shot VideoQA.

2. **Video Encoder Pre-training:** The video is represented as a sequence of frames, each processed separately through a visual backbone. This backbone is pre-trained to map images to text descriptions using a contrastive loss on 400M image-text pairs scraped from the web. The backbone remains static throughout the experiments.

3. **Integrating the Static Language and Vision Components:** The video features are integrated into the language model as a prompt. This prompt is obtained by linearly mapping the visual features to the text token embedding space via a visual-to-text projection. The prompt is then combined with the text embeddings before being forwarded to the transformer encoder that models joint visual-linguistic interactions. To learn powerful multi-modal interactions while keeping the transformer encoder weights static, the transformer encoder is equipped with lightweight adapter modules.

4. **Cross-modal Training:** The newly added modules are trained for the VideoQA task using only readily-available video-caption pairs scraped from the Web. The weights of the pre-trained BiLM and pre-trained visual backbone are kept static. The parameters of the visual-to-text projection module and the adapter modules are trained from scratch.

5. **Adapting to Downstream Tasks:** After training, the model is capable of filling gaps in the input text given an input video along with left and right textual context as part of the input text. The model is applied out of the box to predict an answer given a question about a video. The video can optionally come with textual subtitles obtained using automatic speech recognition.

6. **Input Prompt Engineering:** The input text prompts for several downstream video-language tasks are designed. Each downstream task is formulated as a masked language modeling problem.

7. **Answer Embedding Module:** For each downstream task, a task-specific answer classification head is used to map the mask token in the input text prompt to an actual answer prediction in the set of possible answers.

8. **Fully-supervised Training:** The model can also be finetuned on datasets that provide manual annotations for the target task. The same parameters are trained while keeping the transformer weights and the answer embedding module static.

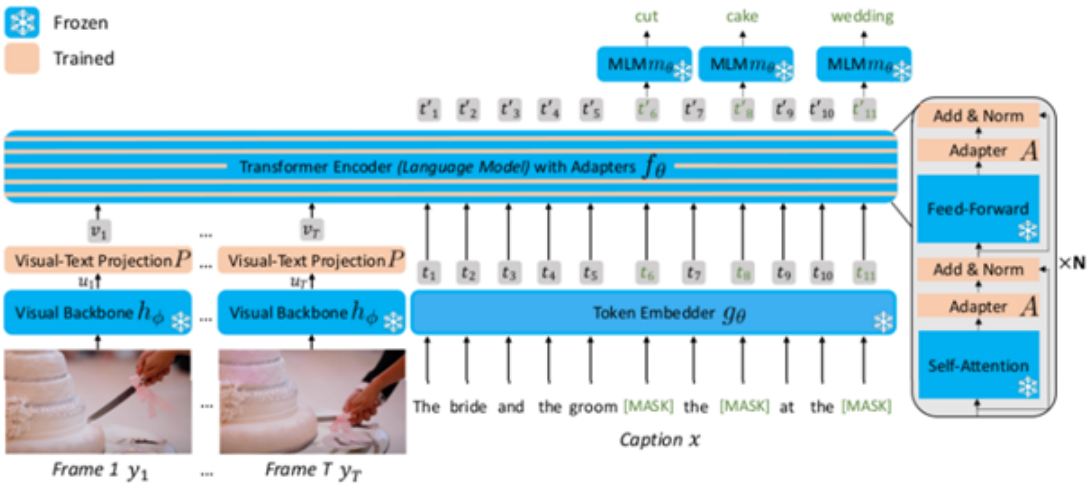


Figure 5: A common framework of VideoQA, consisting of four components: video encoder, question encoder, cross-modal interaction, and answer decoder

The method has demonstrated its ability to enhance the state-of-the-art zero-shot VideoQA on various datasets, performs competitively in fully-supervised settings, and exhibits strong performance in the few-shot VideoQA setting.

2.2 Image Question Answering

In IQA the work is more promising than in VQA we have read multiple papers and chosen the ones that may be candidates in our work. The problem itself is similar to VQA but deals with a single image instead of dealing with a video.

2.2.1 Datasets

1-VQA2

Visual Question Answering (VQA) v2.0 by goyal2017making is a dataset containing open-ended questions about images. These questions require an understanding of vision, language, and common sense knowledge to answer. It is the second version of the VQA dataset.

- 265,016 images (COCO and abstract scenes)
- At least 3 questions (5.4 questions on average) per image
- 10 ground truth answers per question
- 3 plausible (but likely incorrect) answers per question
- Automatic evaluation metric

Figure 6: Sample photos from the VQA2 dataset

The first version of the dataset was released in October 2015. **2-VizWiz** [Gurari et al.] Propose "an artificial intelligence challenge to design algorithms that answer visual questions asked by people who are blind.

For this purpose, we introduce the visual question answering (VQA) dataset coming from this population, which we call VizWiz-VQA. It originates from a natural visual question-answering

setting where blind people each took an image and recorded a spoken question about it, together with 10 crowd-sourced answers per visual question. Our proposed challenge addresses the following two tasks for this dataset: predict the answer to a visual question and (2) predict whether a visual question cannot be answered. Ultimately, we hope this work will educate more people about the technological needs of blind people while providing an exciting new opportunity for researchers to develop assistive technologies that eliminate accessibility barriers for blind people."

- 20,523 training image/question pairs
- 205,230 training answer/answer confidence pairs
- 4,319 validation image/question pairs
- 43,190 validation answer/answer confidence pairs
- 8,000 test image/question pairs

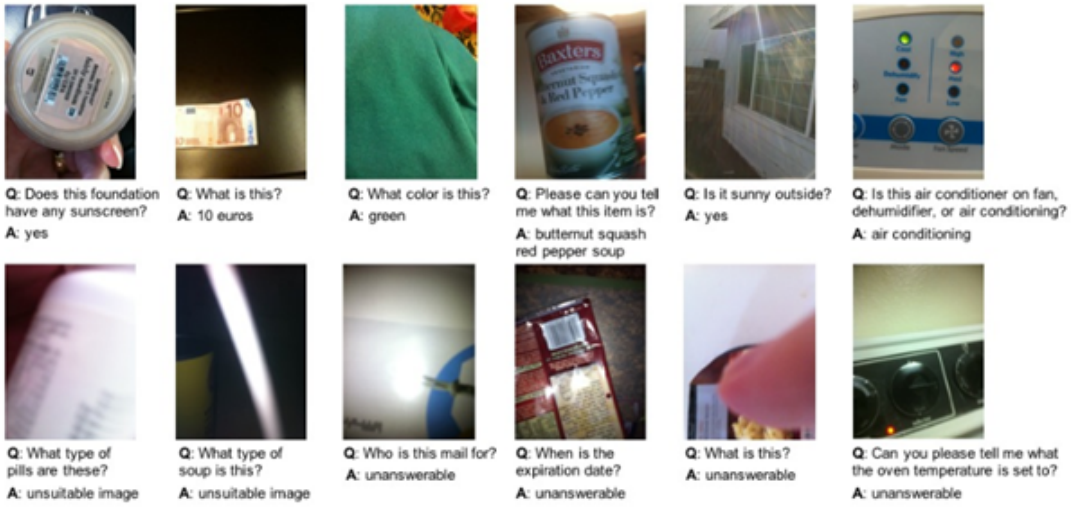


Figure 7: Sample photos from the VizWiz dataset

2.2.2 Techniques and Methods

CLIP

CLIP (Contrastive Language-Image Pretraining) is an innovative neural network developed by OpenAI. It represents a significant breakthrough in the field of artificial intelligence because of its ability to learn from a combination of modalities – specifically images and text.

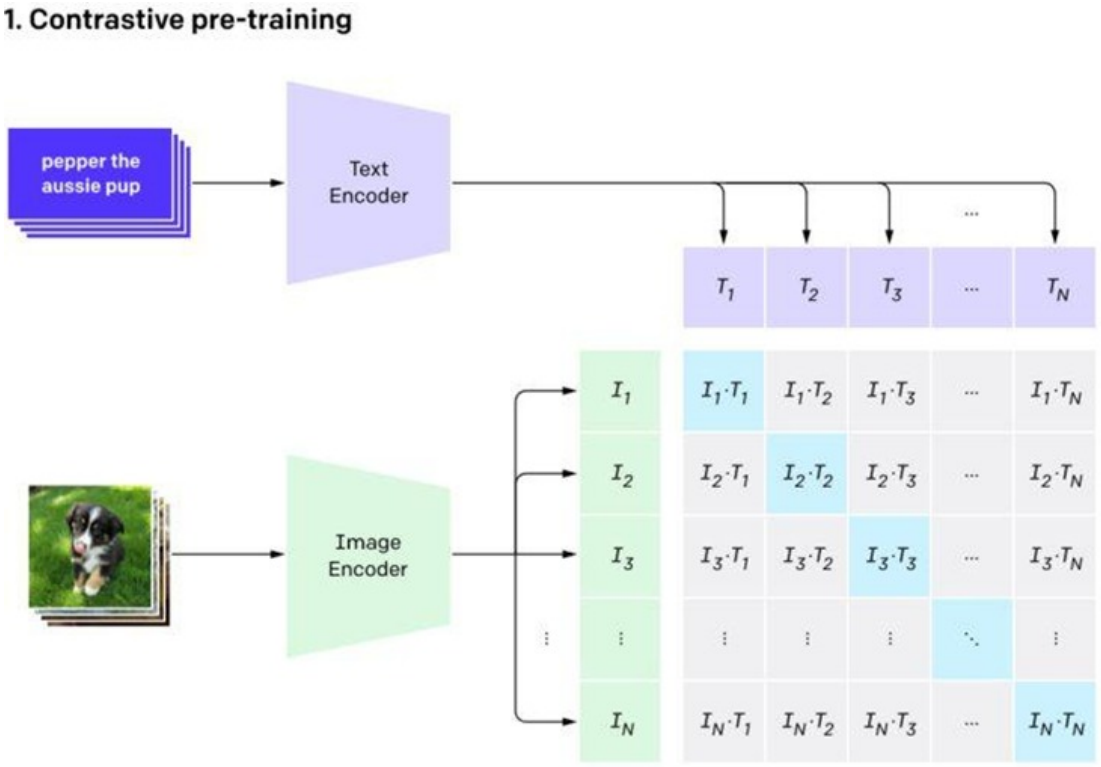


Figure 8: CLIP Training

This multimodal model is trained using a vast dataset comprising 400 million image-text pairs. In each pair, the image and its corresponding text or caption provide contextual information that helps the model understand the content and the relation between the two. This vast training dataset enables CLIP to accurately predict the most relevant text snippet for any given image, showcasing its strong capability in zero-shot learning.

Zero-shot learning refers to a model's ability to correctly infer or predict information about a class that it has never seen during the training process. This is a powerful attribute that models like CLIP, GPT-2, and GPT-3 possess. For instance, even if an image substantially differs from the images used in training, CLIP is likely to produce a suitable caption for it.

The unique strength of CLIP lies in its ability to bridge the realms of computer vision and natural language processing (NLP). It's not only capable of understanding and interpreting images (a field known as computer vision) but also of making sense of human language (the primary focus of NLP).

CLIP consists of two primary components: a text encoder and an image encoder. Both are designed to transform the input data (either text or images) into a series of numbers, or feature vectors, in a shared mathematical space. The process of transformation is known as embedding.

The model's primary goal during the training process is to maximize the "goodness" – in other words, the similarity between feature vectors of matching image-text pairs. Simultaneously, it seeks to minimize the "badness" or the similarity between the feature vectors of non-matching pairs. This is achieved by calculating the cosine similarity between vectors.

In this context, cosine similarity is a metric that measures the cosine of the angle between two vectors. It can range from -1 to 1, where a higher value indicates a greater degree of similarity. Hence, the "goodness" of the model is gauged by how close the cosine similarity of matching pairs is to 1, while "badness" is evaluated by how close the cosine similarity of non-matching pairs is to -1.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

Figure 9: Cosine Similarity Equation

The image encoder in CLIP uses either a ResNet (a deep convolutional neural network) or a Vision Transformer (a model based on transformer architecture) to process the images. Text is processed using a Transformer, which is an NLP model known for its effectiveness in processing sequences of data.

After training, CLIP is capable of several tasks. Given an image, it can predict a fitting caption. Provided with a piece of text, it can find the most relevant image from a database. It can even classify images based on a context described in a sentence or phrase – a task known as zero-shot classification.

To sum up, CLIP is a remarkably advanced model that leverages a wealth of image-text pairs to gain an understanding of both visual and textual data. Its power lies in its ability to map these two distinct types of data onto a shared mathematical space and use the relationships therein to make accurate predictions.

3 Methodology

The methodology employed in this study reflects a dual-faceted approach, uniquely designed to cater to the intricacies and requirements of both image and video question answering tasks. The overall strategy, processes, and techniques are explained in two distinct sections that constitute the core of our methodology: one focused on image question answering and the other on video question answering.

By dividing our methodology into two comprehensive sections, we aim to provide a detailed insight into the research procedures followed, the data handling strategies applied, and the machine learning techniques utilized in both domains. This structured approach allows us to address the complexities and nuances of each modality, thereby providing a comprehensive understanding of our methodology. Both image and video question answering represent complex challenges at the intersection of computer vision and natural language processing. In the following sections, we aim to elucidate our strategies to tackle these challenges, aiming to contribute to ongoing research and potential future applications in these fascinating fields.

3.1 Video Question Answering

The video model methodology used to answer a question based on a video depends on a model steering framework called Composing Ensembles of Pre-trained models via Iterative Consensus (PIC). The framework tries to use the power of pre-trained large language models to generate English text like the GPT family of models, but the LM can not have the ability to interpret visual inputs. We used pre-trained models as generators or scorers and compose them via closed-loop iterative consensus optimization.

In our work, we have 2 different types of models:

Language models In recent years, LMs have improved significantly and are getting closer to AI-complete capabilities, including broad external knowledge and solving a wide variety of tasks with limited supervision. A Transformer based LM typically models interactions between the generated token and past tokens at each time step. Recall that the transformer block has three embedding functions K, Q, V. The first two, K, Q, learn the token interactions that determine the distribution over V. The attention mechanism pools values based on the similarity between queries and keys. Specifically, the pooled value for each token i depends on the query associated with this token Q_i, which is computed using the function Q over the current embedding of this token. The result is obtained as the weighted average of the value vectors, based on the cosine similarity between Q_i and the keys associated with all tokens K_j .

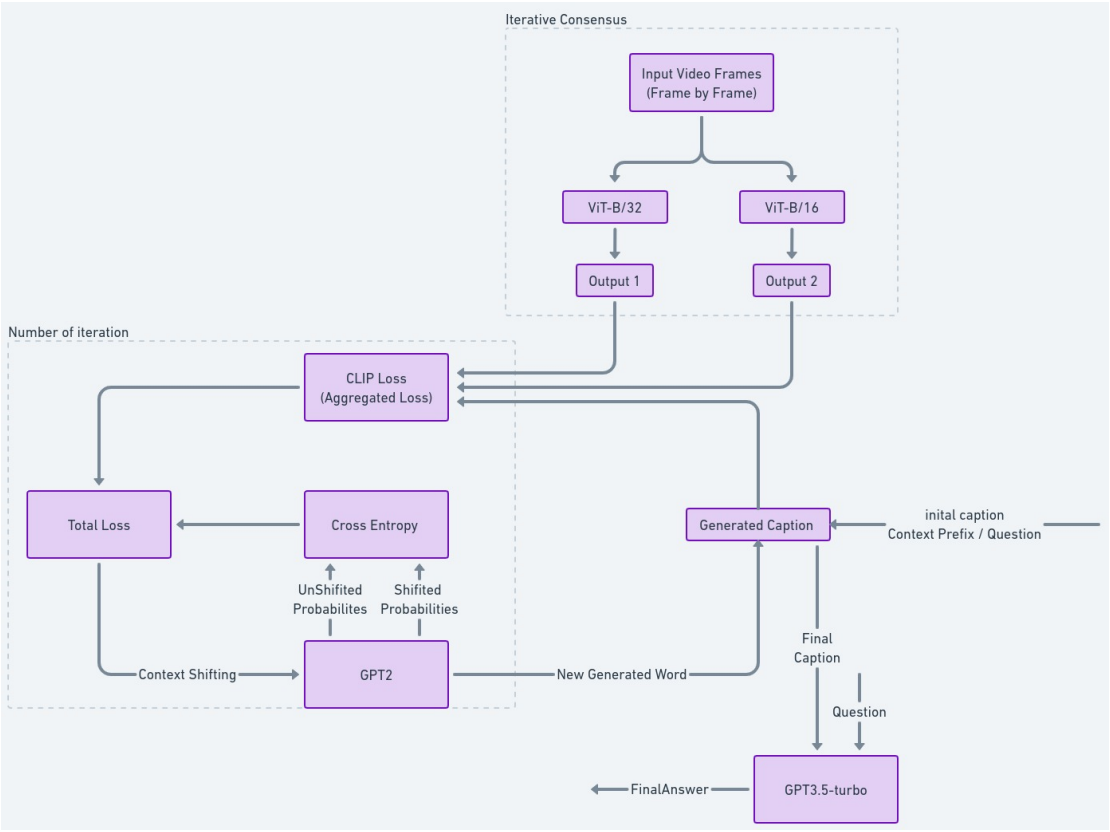


Figure 10: PIC Framework

While K and V are functions, the obtained key and values K_j and V_j are used repeatedly when generating text, one word at a time. K_j and V_j can therefore be stored in what is called a context cache, in order to keep track of past embedding outputs of K and V. The sequence generation process can then be written as

$$x_{i+1} = LM(x_i, [(K_j^l, V_j^l)]_{j<i, 1\leq l\leq L})$$

where X_i is the i-th word of the generated sentence, K_j V_j are the context transformer's key and value of the j-th token, and l indicates the index of the transformer layers, out of a total of L layers. Our method employs GPT-2, which has L = 24 layers. We next describe how we align our LM with the input frame. We do so by modifying, during inference, the values of the context cache leaving the LM unchanged.

$$C_i = \left[(K_j^l, V_j^l)\right]_{j<i, 1\leq l\leq L}$$

As mentioned in the framework, the video model generates captions for videos without considering the associated question. Consequently, the captions may not provide sufficient context for the GPT3.5 model to effectively answer the question. To address this issue, we introduced a solution where the question posed by the user is used as a caption prefix for each frame in the model. This approach enables the GPT2 model and the CLIP model to generate captions that

are more aligned with the given question and facilitate better question-answering capabilities.

CLIP-Guided language modeling Our goal is to guide the LM towards a desired visual direction with each generation step. The guidance we propose has two primary goals: (i) alignment with the given image; and (ii) maintaining language attributes. The first goal is obtained through CLIP, which is used to assess the relatedness of a token to an image and adjust the model (or, rather, the cache) accordingly. For the second goal, we regularize the objective to be similar to the original target output, i.e., before it was modified.

The solved optimization problem adjusts the context cache C_i at each time point and is formally defined as

$$\arg\min_{C_i} \left[L_{CLIP}\left(LM(x_i, C_i), I\right) + \lambda L_{CE}\left(LM(x_i, C_i), \hat{x}_{i+1}\right) \right]$$

where $\hat{x}_{i+1}$ is the token distribution obtained using the original, unmodified, context cache. The second term employs CE loss to ensure that the probability distribution across words with the modified context is close to the one of the original LM. The hyperparameter λ balances the two loss terms. It was set to 0.2 early on in the development process and was unmodified since.

To update C_t, we first use G to generate a set of candidate words $\hat{X}_{t+1} = \{\hat{x}_{t+1}\}$, and then use the feature distance (after softmax) between each sentence (the concatenation of previous words and each new word $\{ x_1, x_2, ..., \hat{x}_{t+1} \}$ where $\hat{x}_{t+1} \in \hat{X}_{t+1}$) and the video frame as the probability of them matching. The CLIP score is the cross-entropy loss L_{CLIP} between this new probability distribution and the original distribution of the next word obtained from the generator G. The gradient of the summed score (multiple CLIP models) is then propagated to G to update C_t :

$$C_t^{k+1} \leftarrow C_t^k + \lambda \nabla C_t^k \sum_{n=1}^{N} L_{CLIP}\left(E_{\theta_n}(x_1, x_2, \ldots, \hat{x}_{t+1}, I)\right)$$

Algorithm 1 PIC framework for video question answering

for each frame in the video **do**

Extract the frame features using clip models (Iterative Consensus)

Set the initial caption as the question itself or any other prefix

for each word (till reaching max sequence length or spacial eos) **do**

for i $\leftarrow$ 1 to $Numberofiterations$ **do**

Calculate CLIP loss between the generated caption so far and the extracted frames features

Shift the context of the GPT2 model using the calculated loss (This is not the actual final shift)

Calculate cross entropy loss between the GPT2 word distribution for the model before and after shifting the context (This act as a regularization term)

Add the Cross entropy loss to the CLIP loss

Shift the model context for the GPT2

end for

Generating a new word using the final shifited distribution

end for

end for

Concatenate all the video frames captions

Pass the caption along the question to GPT3.5-turbo to obtain the final results

3.2 Image Question Answering

This section explores the methodology we followed that combines bootstrapping and model training to enhance the process of generating accurate and contextually relevant image captions. The methodology is based on the Bidirectional Language Image Processing (BLIP) architecture, which serves as the foundation for the entire process.

The text provides a comprehensive explanation of the theoretical foundations behind the methodology, including the training process, loss functions, and the architecture of the BLIP model. Overall, this methodology presents a comprehensive approach to generating high-quality image captions, incorporating theoretical foundations, architectural components, and experimental procedures. .

3.2.1 Theory

Our methodology encompasses bootstrapping and model training. In this section, we unpack the theoretical foundations of both approaches to provide a cohesive explanation of the entire process. Our experiment commenced with the use of the Bidirectional Language Image Processing (BLIP) architecture, which serves as the cornerstone of our methodology. The initial step involves acquiring a dataset of image-text pairs, or "noisy data," from either the internet or an existing database.

We utilized a fine-tuned checkpoint from the COCO dataset as a springboard for training the BLIP architecture.https://www.overleaf.com/project/64a159415378d30f1a6ea8e2 We initially trained the BLIP model on our VizWiz training dataset. To optimize the training process, we devised a technique to create a new row, which extracts the most frequent answer from the top 10 answers. Additionally, we filtered out images for which the maximum answer was labeled as unanswerable. The next stage involved preparing the dataset using a data loader, converting data into tensors, resizing images, and employing a text processor for caption processing.

The BLIP Pretrain class, integral to the BLIP architecture, is responsible for pretraining visual and textual representations. This class integrates a visual encoder based on a Vision Transformer, and a text encoder based on a BERT model. When given an image and corresponding caption as input, the model processes them separately to obtain visual and textual embeddings. The model incorporates several tasks including image-text alignment, image-text matching, and language modeling, with the principal goal of learning representations that align images and captions semantically, accurately match images with their corresponding text descriptions, and generate linguistically coherent text. We achieve this by leveraging three main loss functions: ITC, ITM, and LM.

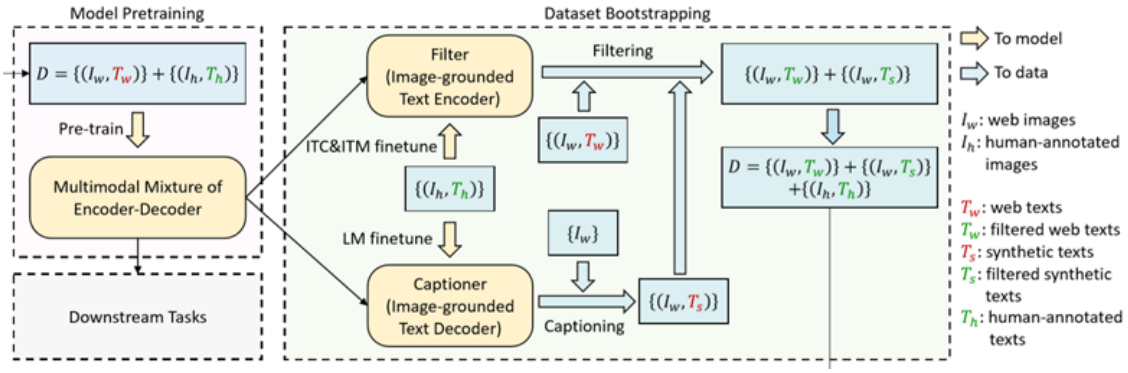


Figure 11: Closer Look Into boostrapping

A specialized module within our model architecture, the captioner, generates captions based on the input image. The language modeling (LM) loss function facilitates this process by measuring how well the generated caption aligns with expected language patterns. This ensures the caption output is not only contextually relevant to the image but also grammatically correct and semantically meaningful. Upon generating captions, we curate a new dataset of image-caption pairs, including both the original and machine-generated captions. We then feed this newly curated dataset into another component of our architecture, the filter.

Algorithm 2 Condensed BLIP framework for multi-modal tasks

for each data instance (image-text pair) **do**

Extract image features and encode text caption

for each word in the caption **do**

Calculate normalized projection of features

for i $\leftarrow$ 1$toNumberofiterations$ **do**

$Updatevisualencoder$

Calculate momentum features

Update feature queue and compute ITA loss

end for

Generate new word in the caption

end for

Forward pass positive and negative pairs through text encoder

Calculate ITM and LM losses

end for

Return ITA, ITM, and LM losses

The filter's function is to assess the compatibility between an image and a caption, providing a score that reflects the caption's descriptive accuracy in relation to the image. A higher score signifies a better match, while a lower score indicates a poorer match. This scoring mechanism plays a critical role in determining the quality of the generated captions, helping us distinguish accurate descriptions from inaccurate ones. Both the caption generation and filtering steps synergistically ensure we produce high-quality and contextually relevant image captions.

Turning to the training theory integral to our methodology, we've previously discussed the general architecture of BLIP and its utility in bootstrapping the VizWiz dataset. Moreover, BLIP serves as a visual question-answering (VQA) model, enhancing its capabilities. In a VQA setting, the model predicts an answer based on an image and a question. Unlike other method-

ologies that frame VQA as a multi-answer classification task, the authors of the BLIP paper made a simple modification to their pre-trained model, leading to a more computationally efficient architecture.

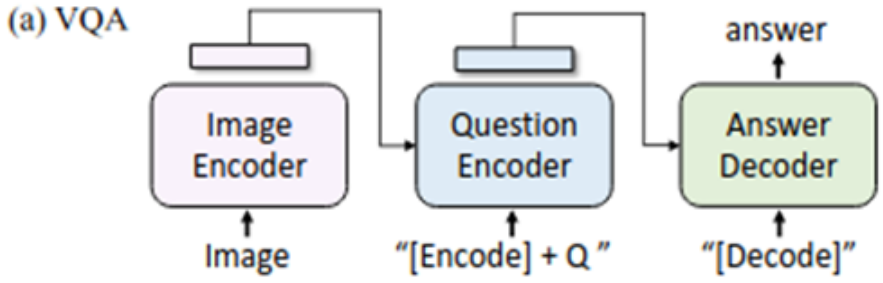


Figure 12: General View of the BlipForQuestionAnswering Model

This modification pertains to the image-grounded text encoder within the Multimodal Mixture of the Encoder-Decoder (MED) model. Each transformer block in the image-grounded text encoder consists of two cross-attention layers to process the two input images. These layers, initialized with the same pre-trained weights, merge their outputs which are then fed to the feed-forward network (FFN). The merge layer performs average pooling in the first 6 layers, and concatenation followed by linear projection in layers 6-12. An MLP classifier is then applied to the output embedding of the [Encode] token. We'll delve further into this architecture in the subsequent architecture section.

For VQA fine-tuning, the authors followed Wang et al.'s approach (2021) and utilized an image resolution of 384 x 384. They carried out experiments using the VQA2.0 dataset, which contains 83k/41k/81k images for training/validation/testing respectively. During their training process, they combined the training and validation splits and incorporated additional samples from the Visual Genome dataset. During the inference phase of VQA, the decoder ranked 3,128 candidate answers. We emulated this fine-tuning approach on our VizWiz dataset, acquiring weights that we later utilized for further fine-tuning. We will present and analyze the results from this process in the following Results and Analysis section.

3.2.2 Architecture

To enhance the image encoder, we adopt the visual transformers approach, breaking down the input image into smaller patches and extracting meaningful features from different regions. By incorporating a special [CLS] token as a representation of the global image feature, our model achieves a comprehensive understanding of the visual content [1].

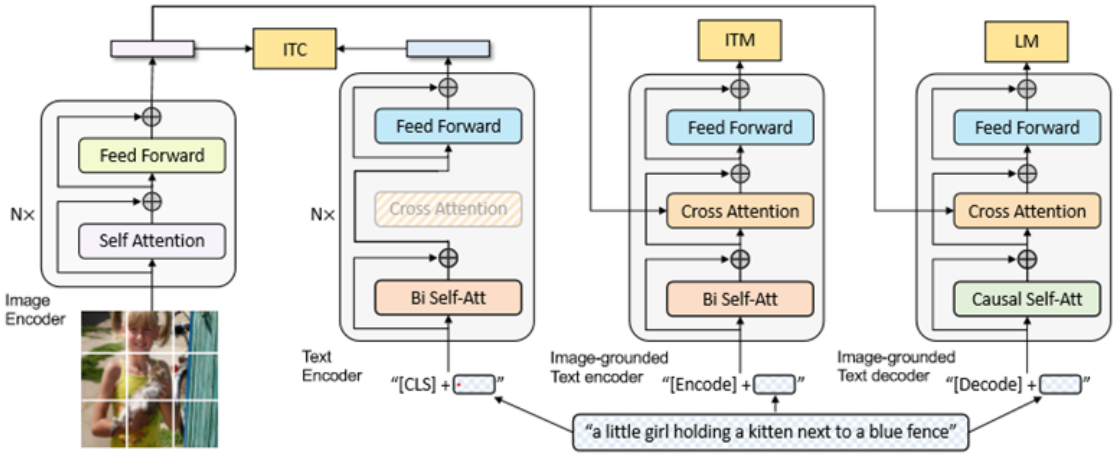


Figure 13: BLIP Architecture

Furthermore, we introduce a cross-attention (CA) layer into each transformer block of the image-grounded text encoder, positioned between the self-attention (SA) layer and the feed-forward network (FFN). This cross-attention mechanism allows the model to attend to relevant visual features during the processing of textual input, facilitating a deeper understanding of the image-text relationship.

For the text encoder functionality, we draw inspiration from BERT [4], adapting its contextualized word representation model for the multimodal context. Our text encoder captures the essence of the input text while considering accompanying visual cues, achieved by incorporating visual information and using the [CLS] token to summarize the sentence.

In the image-grounded text decoder, we make modifications to enable autoregressive text generation based on the combined image and text inputs. We substitute bidirectional self-attention layers with causal self-attention layers, thereby ensuring each generated token solely depends on preceding tokens. This modification enables coherent and contextually relevant text generation. The resulting architecture is highly efficient across various tasks by effectively harnessing both visual and textual information.

Furthermore, we present the specifics of the training model architecture, specifically focusing on the BlipForQuestionAnswering variant. The vision encoder (BlipVisionModel) adopts a Conv2d layer for extracting a sequence of local image feature vectors, enabling detailed analysis of the image's objects and attributes. The BlipEncoder, consisting of BlipEncoderLayer modules, applies attention mechanisms to these features, allowing the model to focus on salient details while suppressing less relevant information. The BlipMLP layer enhances the modeling of complex patterns in the image features.

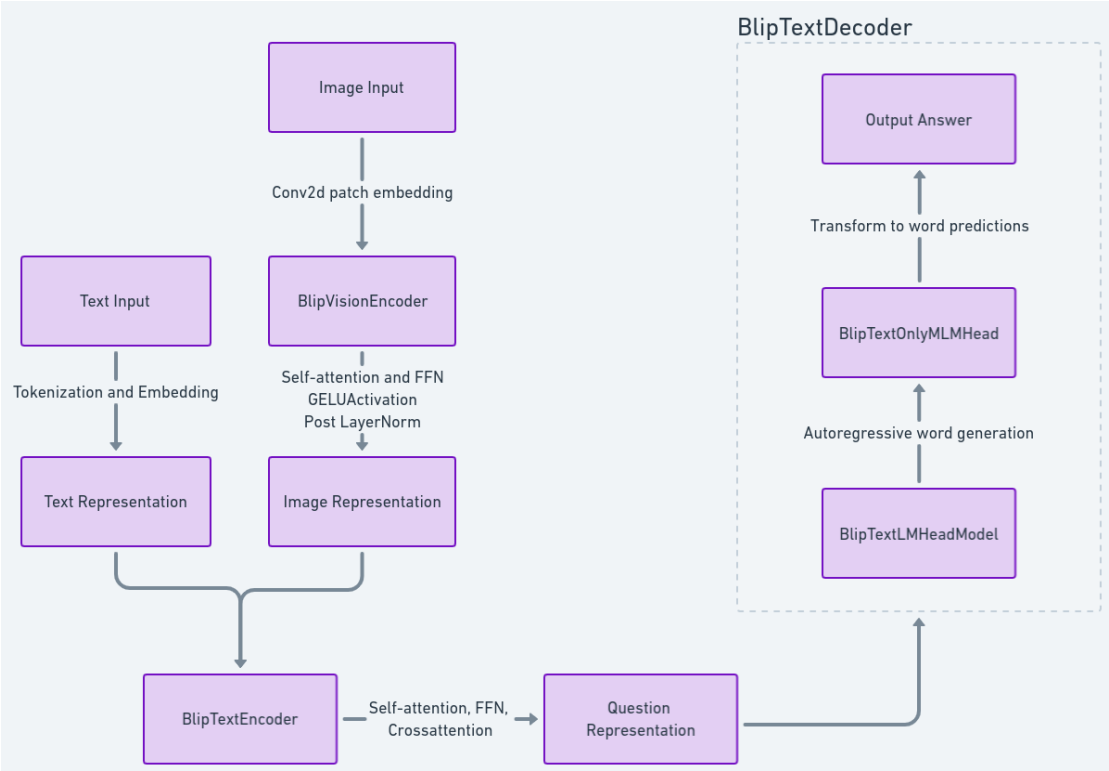


Figure 14: Detailed View of the BlipForQuestionAnswering Model

The text encoder (BlipTextModel) includes the BlipTextEmbeddings layer, which converts words from the input question into vectors using an Embedding layer. Position embeddings encode the word order, preserving the sequential nature of the sentence. Similar to the vision encoder, the BlipTextEncoder contains BlipTextLayer modules, with a BlipTextAttention layer for self-attention and intermediate/output layers for processing the extracted question features.

In the text decoder (BlipTextLMHeadModel), the context from the image encoder and question encoder is utilized to generate the appropriate answer. This decoder structure resembles the encoder, featuring self-attention, intermediate, and output layers. Additionally, the predictions sub-component within the BlipTextOnlyMLMHead incorporates a transform layer to adjust feature dimensions and a Linear layer for the final output, predicting words from the given vocabulary. During training, the BlipForQuestionAnswering model learns from trios of images, associated questions, and correct answers.

It employs supervised learning with a Cross-Entropy loss function to enhance the alignment between the model's predictions and the ground truth. In the inference phase, the trained model receives a new image and associated question, processing the image through the vision encoder, the question through the text encoder, and generating an appropriate answer using the text decoder.

3.2.3 Experiment

As we have talked about the theoretical part of our methodology and the architecture we used in each part, below we will outline our two experiments.

Experiment 1

For the first experiment, the researcher focused on training a model using the VizWiz dataset. The dataset was initially loaded, and the training set was filtered to remove images with unanswerable questions. This was done because it was observed that including unanswerable questions negatively affected the training accuracy. Additionally, a new column was added to the dataset to store the maximum answer for each image.

We utilized the BLIP-Pretrain class from the GitHub repository and made some adaptations to it. One of the changes made was to modify the code to run on a single GPU instead of multiple distributed GPUs. The base model was loaded with fine-tuned checkpoints on the Common Objects in Context (COCO) dataset.

A data loader was defined to pass both the image and the caption to the model. The caption was treated as text and tokenized using a processor. The training was conducted with a batch size of 64 on a Tesla A100 GPU. The model was trained for 5 epochs, and the weights were saved after each epoch. The Adam optimizer with a learning rate of 0.0005 was used for optimization.

After training, the weights from the last epoch were used. The Blip-decoder class was employed with an image size of 224 to generate captions for all the photos in the training set. The generated captions were saved as a text file.

Experiment 2

The second experiment was conducted using the Google Colaboratory, an online Python coding environment that provides hardware support, including GPUs. The initial setup involved mounting Google Drive to access the necessary data.

The VizWiz dataset was loaded using the load-dataset function from the Datasets library. The dataset was divided into training, testing, and validation sets. The questions labeled as 'unanswerable' were filtered out from the dataset. Additionally, a new column called 'max-answer' was added to each instance of the dataset, representing the most common answer to the question. To enhance the training set, the training and validation sets were concatenated and shuffled. A subset of the training set, modified with the most common answers, was also added to the final training dataset.

A model instance of BlipForQuestionAnswering was initialized using a base model provided by Salesforce. An instance of AutoProcessor was also created using the same base model. The AdamW optimizer with a learning rate of 1e-5 and weight decay of 0.01 was utilized. The learning rate scheduler applied was ReduceLROnPlateau, which adjusts the learning rate based on a specified metric.

The model was trained for 5 epochs. Each instance of the training dataset was passed through the model in both the forward and backward directions. The training loss was computed, and optimization was performed. For validation, the model was switched to evaluation mode, and the forward pass was performed without calculating gradients. The validation loss was computed, and early stopping was implemented based on the validation loss. The model weights were saved whenever the validation loss decreased.

After training, the model and processor were saved locally and pushed to a Hugging Face repository specifically created for this purpose. During inference, the pre-trained and trained models, along with their processors, were loaded from their respective paths. The loaded models were used to generate predictions from the validation set. However, the test set couldn't be used to calculate accuracy because the original test set did not provide target answers.

The generated answers were compared against the ground truth answers for evaluation. Accuracy was determined by calculating the cosine similarity between the embeddings of the predicted and candidate answers using the Universal Sentence Encoder. The accuracy was calculated for the validation sets, both for the pre-trained model and the fine-tuned model.

4 Results and Analysis

We will discuss the results we have obtained after conducting our experiments, we will first talk about the results of the Video Question Answering then we follow up on the results of the Image Question Answering

4.1 Video Question Answering

The PIC framework described earlier showed some good performance and other limitations like captioning some irrelevant content to the video and as a result, providing the LM-model with non-accurate information. In this section, we will show the result of the PIC model on a few videos.

The first example is for a video for the Burj-khalifa located in Dubai City the model takes a question asking for its location and provides a correct answer.

Figure 15: Question: In which city the tower is located
Answer: The tower is located in Dubai

Another example that shows some limitations for the model is the Saudi Arabia football player who was wearing a green shirt in the world cup match against Argentina the model output that the player was wearing a red and green shirt not only a green shirt

Figure 16: Question: What color is the shirt of the player who scored the goal?
Answer: the shirt is green and red

The captioning mechanism in the PIC framework captions the video independently in the caption, which might result in a caption that is irrelevant to the question and the LM won't be able to provide a clear answer. To overcome such a problem we used the question being asked as a caption prefix for the model, which helped the model to make better captions and produce better answers.

Figure 17: Question: What are the ingredients used in the pizza
Answer without question prefix: Unfortunately, the provided caption does not mention the ingredients used in the pizza
Answer with the question prefix: The ingredients used in the pizza are fresh mozzarella and flour

4.2 Image Question Asnwering

The metric we used is accuracy, we calculated it according to the way the VizWiz dataset creators specified, below is the equation they used.

$$acc = \min\left(\frac{\#Humansthatansweredthatanswer}{3}, 1\right) \tag{2}$$

This Is how we went about implementing that equation. The accuracy was calculated using the Universal Sentence Encoder from TensorFlow Hub. The embeddings of the predicted and candidate answers were generated, and the cosine similarity was calculated. A predicted answer was considered correct if its maximum similarity score with the candidate answers was more than or equal to 0.8. The final accuracy was the average of all the predicted answers' accuracies.

The experimental results, including both training and validation losses over five epochs and accuracies, are summarized in the following table:

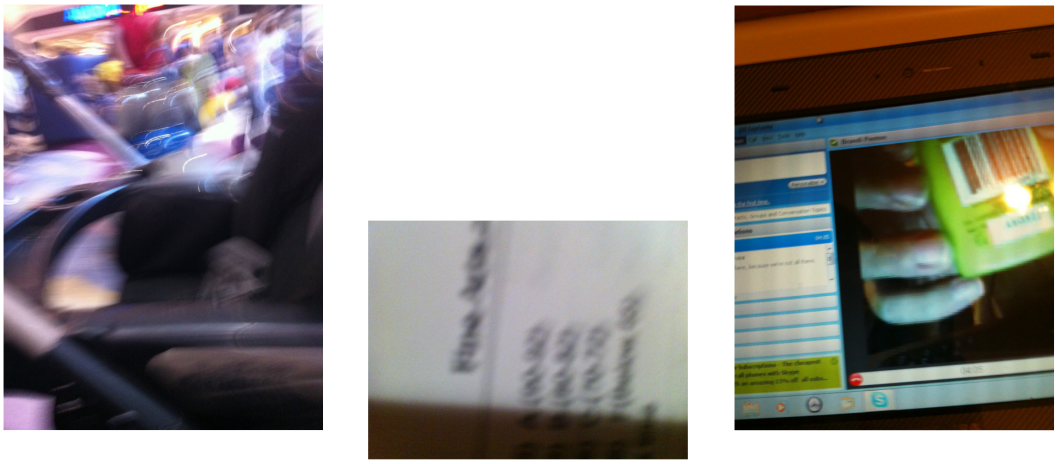
Epoch	Training Loss	Validation Loss
1	0.29629	2.13579
2	0.23097	1.78319
3	0.19797	1.64771
4	0.17895	1.59186
5	0.17059	1.56113

Table 1: Training, Validation Loss and Accuracy Over Five Epochs

Model	Base Model	Trained Model
Accuracy	33.69%	87.30%

Table 2: Trained and Base Model Accuracy on the VizWiz Validation-Set

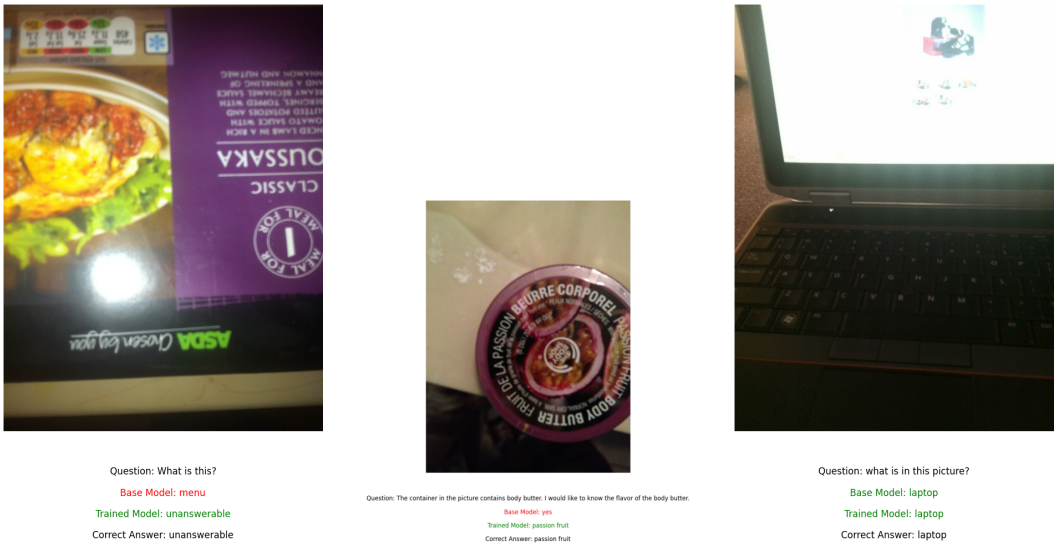
The presented results demonstrate a subset of the generated dataset that we used in addition to the original VizWiz to train the BLIPForQuestionAnswering



(a) Example 1 (b) Example 2 (c) Example 3

Figure 18: Sample of the generated dataset that was used in the training

Here are some example pictures from the dataset and other pictures we got online and the Base, Trained model predictions on them



(a) Example 1 (b) Example 2 (c) Example 3

Figure 19: Base and Trained Model predictions on samples from the VizWiz validation-set

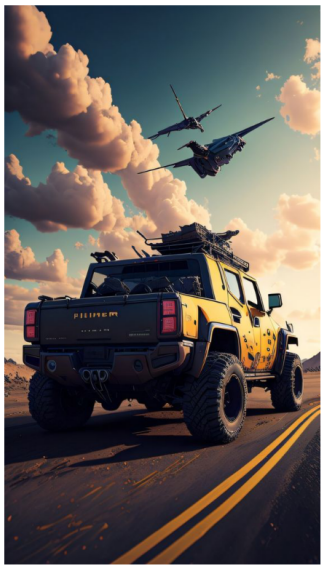
Question: How many cats are in the picture

Base Model: 2

Trained Model: 3

Correct Answer: 2

(a) Example 1



Question: what vehicles are in the picture

Base Model: jeep and jeep

Trained Model: jeep

Correct Answer: car and two planes

(b) Example 2

Question: How many cars are in the picture?
Base Model: 3
Trained Model: 3
Correct Answer: 3

(c) Example 3

Figure 20: Base and Trained Model predictions on samples from the Internet

The model has shown a consistent reduction in the loss metric over subsequent epochs, suggesting that it is learning and adapting to the data. The accuracy of the trained model in the last epoch was significantly higher than the baseline model (33.69%), which indicates a successful training process.

But if you look closely at the samples provided above, you will see that the trained model performs better on the pictures from the VizWiz dataset and doesn't perform as bad as the base model on images from the internet, this indicates that the model we developed has learned the kind of images the VizWiz was created from and it could help in solving the problem of aiding visually impaired people.

5 Demo

Every major research group offers demos of their state-of-the-art models, so we have developed a simple web application to provide similar demos for our own state-of-the-art models. These models were considered among the best when their papers were published and are still among the top 5 models in their respective fields.

Our main source of inspiration for this web application was the Hugging Face website, which provides demos for a wide range of transformer-based models and is also responsible for the popular transformers library that we extensively used in building our models.

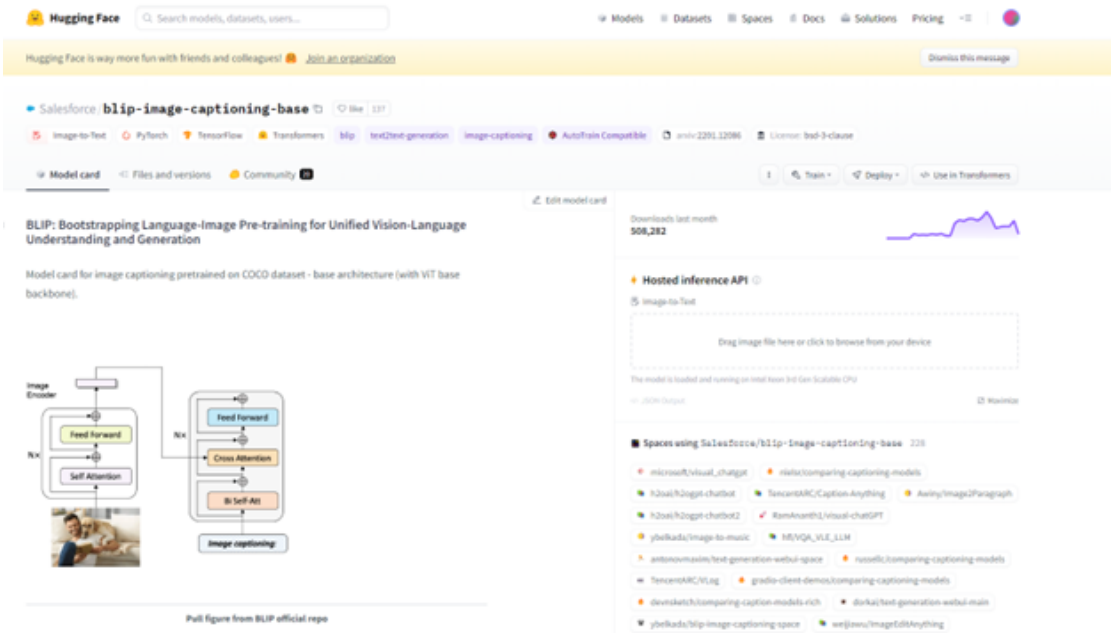


Figure 21: Hugging Face

5.1 Back end

Given that most AI libraries are written in Python, we needed to build a web app in Python as well. After conducting some research, we came across two web frameworks: Django and Flask. Django is a comprehensive framework that handles both the front-end and back-end of the application, offering extensive customization options. On the other hand, Flask is a more versatile and beginner-friendly framework that provides us with a simple way to pass data from the front end to the back end, where the model's inference takes place. This allows us to generate answers and send them back to the front end. Hence, we chose Flask as it fulfilled our specific requirements.

5.2 Front end

For the front end, we utilized Bootstrap v5 to create navigation bars, forms, buttons, and other styles. Since Flask interacts with the front end using Jinja, it was straightforward to apply Bootstrap directly to the HTML files without the need for additional back-end modifications. This approach helped keep our codebase clean and organized.

5.3 Our Web App

Our web app is a very easy-to-use application with only 3 pages a landing page and 2 pages for the two models the video and image model. The landing page or the home page contains only the introduction to our web app and project and contains a little brief about the video and image models with hyperlinks that if clicked on will transfer automatically to the image or video page.

Figure 22: Glimpse at Our Landing Page

For both the image and video question answering pages, it's divided in half as described above a half for the form that takes the model inputs and a half for a description of the model.

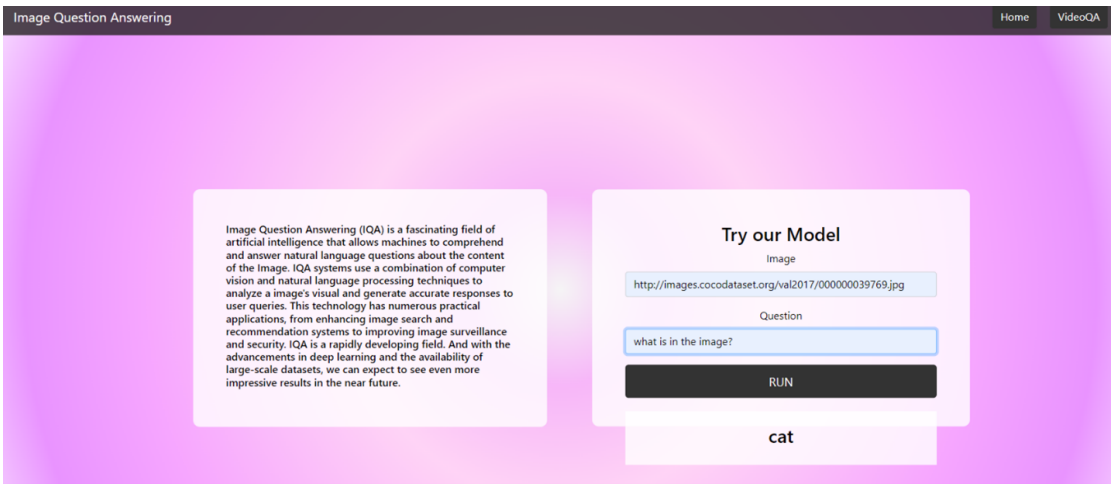


Figure 23: PIC Framework

In this screenshot, there is a question and a URL given to run the blip image model with our

trained weights. In the picture, there were two cats sitting on a sofa or a bed.

To deploy our web app and run it with our resource-intensive models, we required a powerful machine. Even our local laptops and machines struggled to load and run both models simultaneously. Consequently, we began searching for a suitable deployment platform. However, we encountered difficulties and ultimately decided to run the Flask app on Google Colab, which provides a powerful GPU (T4) with 16GB of VRAM and 12GB of RAM.

Flask runs on the Google Colab virtual machine as its local host, and a mirror link is provided by the Google Colab library for interacting with the local host. However, direct access to the local host is restricted due to Colab limitations and Google's security terms. We are also limited by the duration of each session, which is 8 hours of GPU running time. Both the image and video question-answering pages follow the same layout as described earlier, with a form for model inputs on one side and a model description on the other.

6 Challenges and Future Work

In this section we will discuss some of the challenges we faced that hindered our progress in this work and we will also share some ways this work can be expanded upon

6.1 Video Question Answering

Video Question Answering is a new field so there are still many things to discover in it and many things to experiment with. Also, the development that happened in the last year in AI generally and in the VQA field specifically due to many factors as the fast development that happened in NLP and the use of LLMs in it that he benchmark on many datasets in VQA have changed multiple times in the timeframe of few months. As a result, we have few insights on the future work that can be added to our work or in the VQA problem generally in addition to some problems that we have faced that may be fixed in future work

6.1.1 Challenges

- **Computational power:** Working with large datasets and LLMs requires a significant amount of computational power that our devices couldn't handle. Initially, we tried using free cloud providers such as Google Colab, but the resources provided were insufficient. To overcome this limitation, we sought funding and through personal relationships, we gained access to Huawei Cloud through a startup called Snappers. We utilized approximately USD 100 worth of GPU resources, but the funding duration ended earlier than anticipated. As a result, we subscribed to Google Colab Pro for USD 50, which allowed us to complete our three models.

- **Irrelevance of frame captions in PIC to the questions:** During the generation process of the PIC model, we encountered issues with the relevance of the generated captions to the given questions. The captions were not sufficient to address the provided questions. To resolve this problem, we modified the model by providing the question as a prefix to the generated caption. This adjustment resulted in improved results and better alignment between the captions and questions.

6.1.2 Future Work

- **Verifying the PIC results:** This was a goal for our project that we did not achieve because the PIC framework proposed originally and deployed by us takes a long time to run even using high computational power. It was such that running it even on a small dataset like ActivityNetQA was not possible. So, it is suggested that anyone who has more computational power or a bigger fund than what we got may verify the results of the PIC model.

- **Working on the Audio Modality:** When we started working on the VQA problem, even multi-modal datasets that were available were used to work on text modality as subtitles or plots. However, more recent work in the last 2 months started working on audio modality as in the VALOR model or VAST model, both developed by [Chen et al.]. Adding Iterative consensus may improve their results with the VALOR or VAST model themselves as the generator in our framework and CLIP model(s) as scorers. Additionally, developing a similar model to the clip that works also with audio modality can be explored.

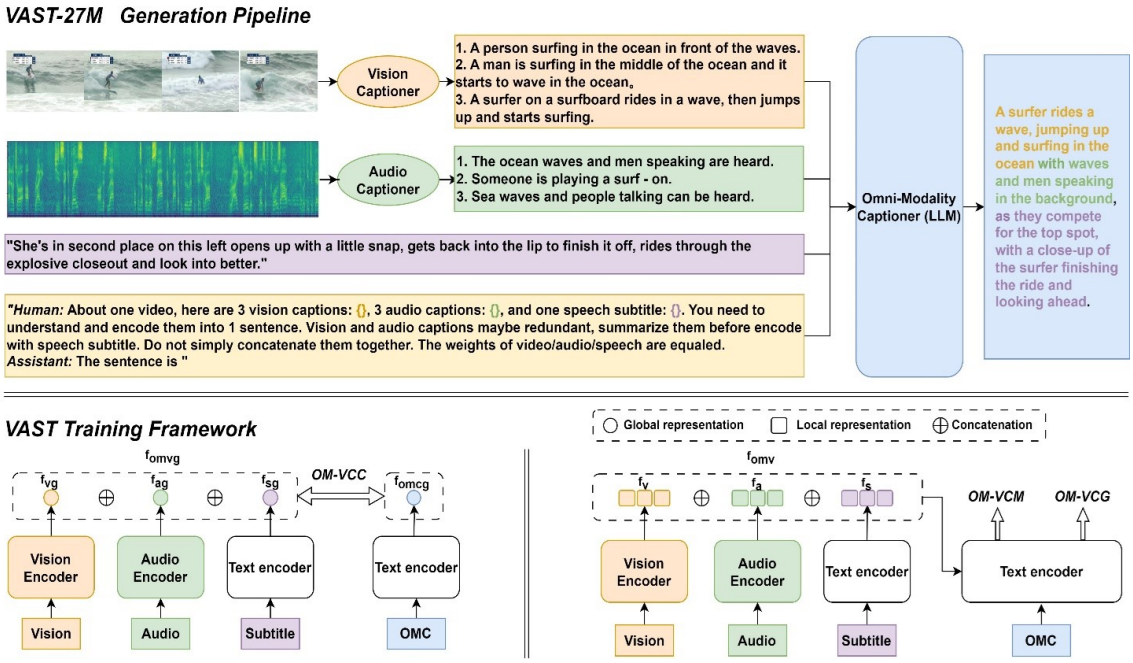


Figure 24: Illustration of VideoQA, Multimodal VideoQA (MM) and Knowledge-Based VideoQA (KB-VQA)

- **Improving the Speed of the PIC framework:** Applying the PIC framework in the way it was proposed is a time-intensive process. It uses results from the inference of multiple models including LLMs, so generating a caption for even 1 frame takes multiple seconds. Changes in the framework may turn the model into something working in real-time as our image model or close to it but with not that long intervals.

- **Adding Visual Reasoning to our model:** Some models, like WildQA by [Castro et al.], have started to work on visual reasoning which provides evidence on why did the model answer this question in that way using Visual Reasoning techniques. Applying this to our model can improve the quality of our model and seems an interesting task to test through.

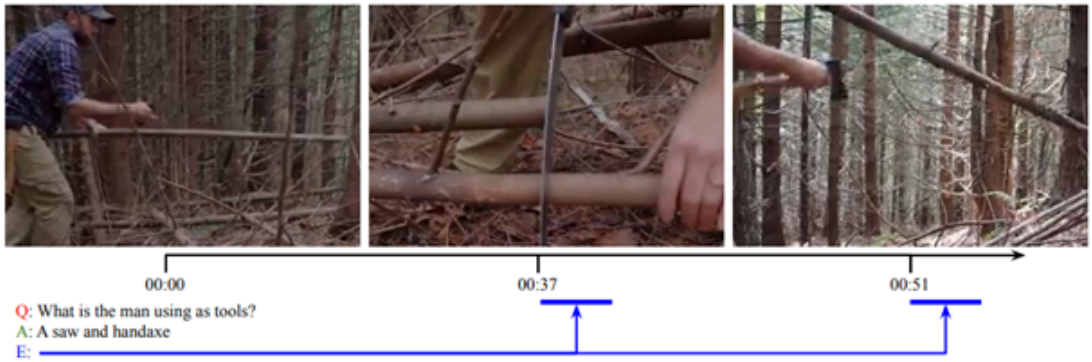


Figure 25: Illustration of VideoQA, Multimodal VideoQA (MM) and Knowledge-Based VideoQA (KB-VQA)

6.2 Image Question Answering

6.2.1 Challenges

- **Computational Power:** The VizWiz dataset is a large-scale dataset that consists of images and their corresponding captions. Training a model like BLIP on such a large dataset requires significant computational resources. This includes not only a powerful GPU for the training process itself but also sufficient storage to hold the dataset and the model's parameters. Moreover, the training process can be time-consuming, potentially taking days to complete. We spent 150 dollars on computational resources from two different sources: Huawei Cloud and Colab Pro Plus.

- **Dataset Nature:** The VizWiz dataset is unique in that it is sourced from blind users and contains many images that are poorly framed, blurred, or have other quality issues. This presents a challenge for any image-based model, as these images may not contain clear or easily identifiable objects. Furthermore, the captions in the VizWiz dataset can be quite diverse and complex, reflecting the wide range of questions and scenarios encountered by blind users in their daily lives. This complexity can make it challenging for the model to learn effective representations for both the image and text data.

- **No Documentation:** Unlike some other popular models, the BLIP model does not come with extensive documentation or pre-existing benchmarks for tasks like other models. We had to adapt the BLIP model for our GPU and our dataset, and it took a lot of time.

- **Utilizing LLMs in a more efficient way** In the last 2 weeks, a paper called Video-LLama by [Zhang et al.] utilizes LLM in a more efficient way as it uses Temporal Dynamics as it includes a Video Q-former that captures temporal changes in visual scenes, which is essential for understanding videos.

  Also, it uses the embedding from Imagebind which produced better results, and Training Data specified to this problem specifically and used the model to make it chat-like. So they proved LLMs can achieve good results and faster than ours relatively so if someone can recreate their experiments with some refinements he would get good results

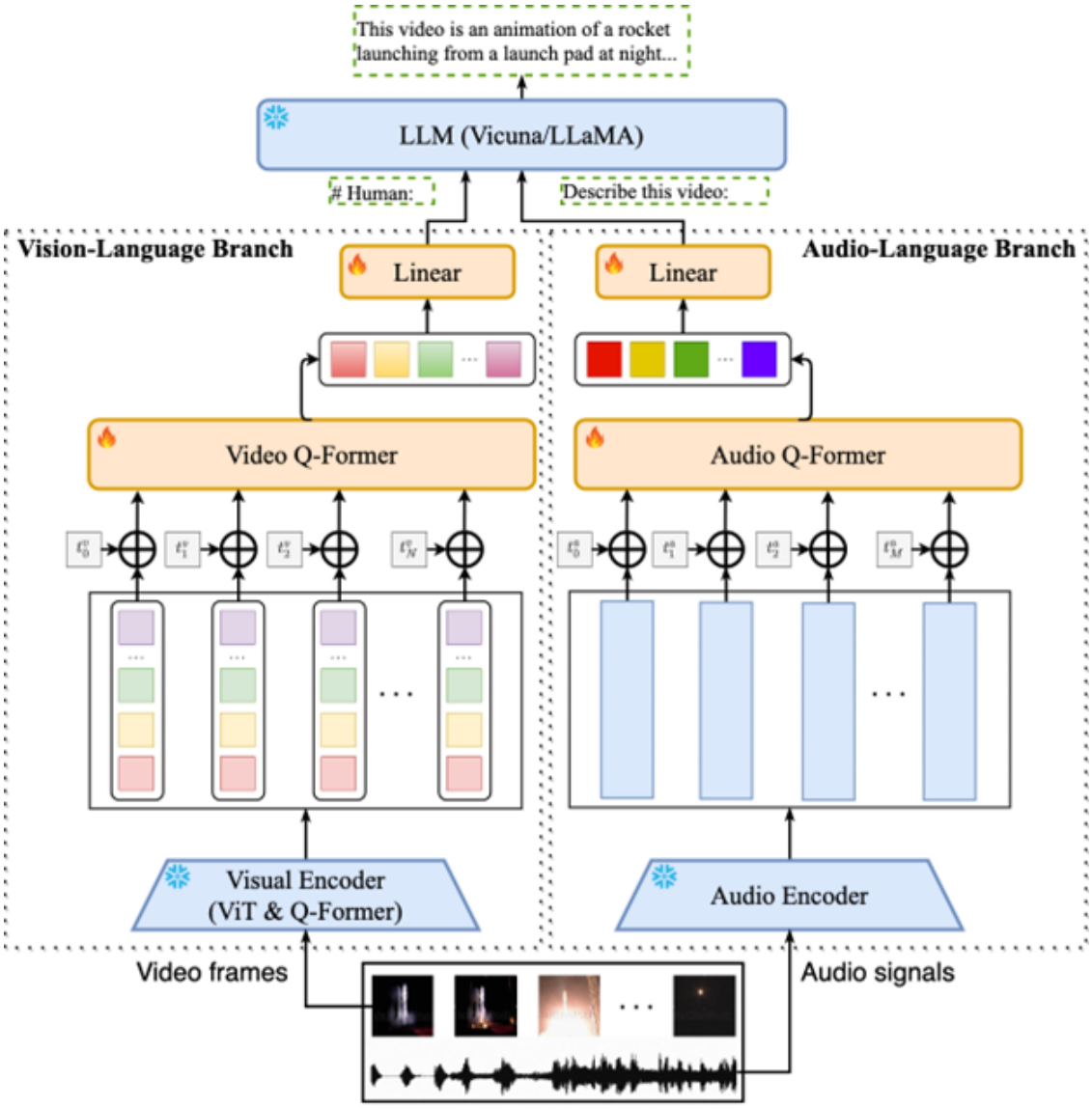


Figure 26: Video LLaMA

6.2.2 Future Work

- **Multimodal Applications:** BLIP could be extended to other multimodal tasks beyond image captioning and filtering. For example, it could be used in video captioning, where the model generates descriptions for video clips.
- **Fine-tuning on Specific Domains:** BLIP could be fine-tuned on specific domains to improve its performance on certain tasks. For example, if you're interested in medical imaging, you could fine-tune BLIP on a dataset of medical images and their corresponding descriptions.
- **Improving Robustness:** Future work could focus on improving the robustness of BLIP. This could involve developing techniques to handle noisy or incorrect captions, such as filtering out captions that are likely to be incorrect or implementing mechanisms to verify the accuracy of generated captions.

7 Conclusion

In this study, we proposed a methodology for visual question-answering tasks, utilizing the BlipForQuestionAnswering model that combines image and text encoders with self-attention mechanisms. However, our experiments faced challenges due to the unique nature of the VizWiz dataset, consisting of images taken by partially impaired individuals, which often resulted in blurred or vague images that were difficult to answer. Consequently, training the captioner on this dataset proved to be exceptionally challenging, compounded by the fact that the training was based on maximum answers, which could be as short as a single word or two.

While the bootstrapping technique employed in the study has shown promising results in previous datasets, our experiments on the VizWiz dataset did not yield the same level of success. However, we were able to generate a new filtered dataset with newly generated captions, which can be considered a valuable outcome.

In summary, our proposed methodology, incorporating the BlipForQuestionAnswering model, demonstrated its effectiveness in generating accurate answers for visual questions. Nonetheless, the limitations of the VizWiz dataset, such as the quality of images and the uniqueness of the answers, posed significant challenges during the training process. Future research should focus on addressing these challenges, exploring alternative approaches that better handle the characteristics of such datasets, and improving the performance of the captioner on challenging images.

Although our experiments may not have achieved optimal results, the methodology presented in this study contributes to the field of visual question answering by integrating image and text encoders with self-attention mechanisms. Our study highlights the importance of dataset characteristics and the need for tailored approaches when dealing with unique and challenging datasets like VizWiz. By acknowledging these limitations and suggesting future directions, this research opens up opportunities for further investigation and improvement in the fascinating field of visual question answering.

Regarding the Video framework it still needs a tremendous amount of effort since the models requires a lot of computational power. The PIC framework requires a lot of computational power, because the model is a composition of four different clip models, nevertheless the new proposed method using LLMs like videoLLaMA is a good area of research that might makes a good impace in the video.

References

Bao, H., Wang, W., Dong, L., Liu, Q., Mohammed, O. K., Aggarwal, K., Som, S., & Wei, F. (2021). Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *arXiv preprint arXiv:2111.02358*.

Barra, S., Bisogni, C., De Marsico, M., & Ricciardi, S. (2021). Visual question answering: Which investigated applications? *Pattern Recognition Letters*, *151*, 325–331.

Castro, S., Deng, N., Huang, P., Burzo, M. G., & Mihalcea, R. (2022). Wildqa: In-the-wild video question answering.

Chen, S., He, X., Guo, L., Zhu, X., Wang, W., Tang, J., & Liu, J. (2023). VALOR: Vision-audio-language omni-perception pretraining model and dataset.

Chen, S., Li, H., Wang, Q., Zhao, Z., Sun, M., Zhu, X., & Liu, J. (2023). VAST: A vision-audio-subtitle-text omni-modality foundation model and dataset.

Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., et al. (2022). Pali: A jointly-scaled multilingual language-image model. *arXiv preprint arXiv:2209.06794*.

Deuser, F., Habel, K., Rösch, P. J., & Oswald, N. (2022). Less is more: Linear layers on clip features as powerful vizwiz model. *arXiv preprint arXiv:2206.05281*.

Floridi, L., & Chiriatti, M. (2020). Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, *30*, 681–694.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., & Parikh, D. (2016). Making the V in VQA matter: Elevating the role of image understanding in visual question answering.

Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., & Parikh, D. (2017). Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 6904–6913.

Gurari, D., Li, Q., Stangl, A. J., Guo, A., Lin, C., Grauman, K., Luo, J., & Bigham, J. P. (2018). Vizwiz grand challenge: Answering visual questions from blind people.

He, P., Liu, X., Gao, J., & Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Hu, Y., Hua, H., Yang, Z., Shi, W., Smith, N. A., & Luo, J. (2022). Promptcap: Prompt-guided task-aware image captioning. *arXiv preprint arXiv:2211.09699*.

Karadeniz, A. S., Erdem, E., & Erdem, A. (2021). Burst photography for learning to enhance extremely dark images. *IEEE Transactions on Image Processing*, *30*, 9372–9385.

Li, J., Li, D., Xiong, C., & Hoi, S. (2022). Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. *International Conference on Machine Learning*, 12888–12900.

Lin, Y., Xie, Y., Chen, D., Xu, Y., Zhu, C., & Yuan, L. (2022). Revive: Regional visual representation matters in knowledge-based visual question answering. *arXiv preprint arXiv:2206.01201*.

Marino, K., Rastegari, M., Farhadi, A., & Mottaghi, R. (2019). Ok-vqa: A visual question answering benchmark requiring external knowledge. *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition*, 3195–3204.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. (2021a). Learning transferable visual models from natural language supervision. *International conference on machine learning*, 8748–8763.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021b). Learning transferable visual models from natural language supervision.

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding lstm–a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Tenney, I., Das, D., & Pavlick, E. (2019). Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

Wang, W., Bao, H., Dong, L., Bjorck, J., Peng, Z., Liu, Q., Aggarwal, K., Mohammed, O. K., Singhal, S., Som, S., et al. (2022). Image as a foreign language: Beit pretraining for all vision and vision-language tasks. *arXiv preprint arXiv:2208.10442*.

Yang, A., Miech, A., Sivic, J., Laptev, I., & Schmid, C. (2022). Zero-shot video question answering via frozen bidirectional language models. *arXiv preprint arXiv:2206.08155*.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., Tay, F. E., Feng, J., & Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. *Proceedings of the IEEE/CVF international conference on computer vision*, 558–567.

Zhang, H., Li, X., & Bing, L. (2023). Video-LLaMA: An instruction-tuned audio-visual language model for video understanding.