Visual Question Answering (VQA) System for Enhanced Understanding of Machine Learning Classes

Radek Holik rholik@mail.yu.edu

Ruslan Gokhman rgokhman@mail.yu.edu

Manish Kumar Thota mthota@mail.yu.edu

Yeshiva University, NYC, NY
Katz School of Science and Health

Abstract

This study embarks on dual objectives in the field of Visual Question Answering (VQA): the construction of a comprehensive dataset from Yeshiva University's machine learning course, and the development of a sophisticated multimodal VQA model. The dataset assembly involved three meticulously executed phases: image and transcript collection, and question-answer pair development. Despite challenges such as the complex nature of the image collection and the lower quality of the source audio material, the effort was successful in laying a robust foundation for the analysis.

Central to the project is the integration of four distinct VQA models: LLaVA 1.5, Salesforce's BLIP, a custom GPT2-based model coupled with Vision Transformer (DeiT), and the Pix2Struct model. These models form the backbone of an advanced multimodal VQA system, designed to enhance the traditional educational experience. The models were rigorously evaluated using key metrics like Cosine Similarity, BLEU Score, and ROUGE Score, ensuring a comprehensive assessment of each model's ability to process and interpret complex educational content.

The results of the study, including the challenges and lessons learned from the dataset construction and comparative analysis of the models, make a significant contribution to the field of VQA. They highlight the complexity and potential of developing systems capable of nuanced understanding and response generation in visually situated speech scenarios, paving the way for future advances in AI and machine learning.

1. Introduction

In the digital age, where visual content is increasingly central to educational environments, there is a growing need for systems capable of comprehensively understanding and interacting with such content. Machine learning (ML) lectures, a critical component of online education, are particularly challenging in this regard. Loaded with complex diagrams, mathematical equations, and dense information, these lectures present unique challenges for comprehension and engagement.

Traditional Visual Question Answering (VQA) systems, which focus primarily on static images, have achieved considerable success in answering questions related to still visual content. However, the complexity and depth of educational material, especially in ML lectures, requires a more advanced level of understanding that goes beyond the capabilities of standard VQA systems.

This study aims to extend the boundaries of VQA to the domain of complex educational content. To this end, we have explored several models and selected the most effective one tailored to our unique dataset. This dataset, which consists of machine learning lectures supplemented with open-ended questions and answers, comes from Yeshiva University, Katz School of Science and Health (YUKSSH) [1]. Our chosen model is characterized by the integration of various external knowledge sources, in-depth analysis of both visual and textual data, and a sophisticated multi-stage reasoning process.

By focusing on the model that best handles the intricacies of educational content, our approach demonstrates superior performance compared to methods designed for more general image content. We believe that this work will significantly improve the understanding of complex educational materials and advance the development of question-answering systems in academic settings.

2. Related Work

Visual Question Answering (VQA) emerged as a research area focused on static images. Malinowski and Fritz [2] are early pioneers who introduced one of the first open-ended datasets for image question answering. They also proposed a recurrent neural network (RNN) model for the VQA task. This dataset became the foundation for subsequent research. Building on this foundation, Zhu et al. [3] incorporated spatial attention mechanisms into VQA models to enable focus on relevant image regions. This combination of vision and attention mechanisms proved to be essential.

The expansion of the VQA field from 2015 to 2017 was greatly accelerated by the availability of benchmark datasets. These datasets provided a consistent methodology for tackling VQA problems, thus fostering a more cohesive research agenda. Notable examples that gained wide community adoption include VQA v1.0[4] and the Microsoft COCO-VQA dataset [5]. These benchmarks consisted of diverse and complex question-answer pairs that challenged the limits of VQA models. In addition to providing a common basis for comparing different models, these benchmarks drew attention to issues that arise in practical contexts, paving the way for further research progress.

In recent years, architectural improvements have significantly advanced image VQA. For example, bottom-up and top-down attention (Anderson et al., [6]) combine high-level semantic and low-level visual features, multimodal transformer networks (Su et al, [7]) jointly model image and text modalities, and graph neural networks (Norcliffe-Brown et al., [8]) incorporate structured knowledge. One notable model is MiniGPT-4 (Zhu et al., [9]), which uses a transformer-based architecture pre-trained on large multimodal databases and achieves new state-of-the-art results on several VQA benchmarks.

The subsequent wave of VQA research aimed to address the dynamic input of video. Initially, VQA datasets were designed for movies and cartoons. Some examples of such datasets include MovieQA (Tapaswi et al., [10]), which is based on plot summaries and question-answer pairs related to movies, and TGIF-QA, an animated dataset collected by Jang et al. [11]. Jang et al. also proposed a two-state recurrent model for video QA. More recent research has prioritized VQA for longer real-world videos, as exemplified by the TV-QA dataset (Lei et al., [12]), which consists of question-answer pairs related to six popular TV shows. This research also developed spatio-temporal VQA models specifically for this dataset.

With the basic aspects of VQA well established, the community moved on to domain-specific challenges. Recently, VQA for educational videos has gained increasing interest, as this domain presents additional challenges requiring complex reasoning skills. Datasets in this genre include EgoVQA (Fan, [13]) with cooking video QA pairs, How2QA (Yi et al., [14]) with science video clips and questions, and VideoQuAD (Zhong et al., [15]) compiled from university machine learning lectures. Such specificity pushes models to be more versatile and adaptable to content variations. Hierarchical recurrent models and multistep reasoning frameworks are among the techniques used so far.

The main architecture of the model is based on BLIP2[16], a generic and efficient pre-training strategy that bootstraps vision-language pre-training from standard frozen pre-trained image encoders and frozen large language models. This pioneering approach is executed in two crucial stages, each of which contributes to the improvement of vision-language understanding. The first phase focuses on the symbiotic evolution of vision and language representations, while the second phase promotes the generation of textual content from visual input.

In the initial stage, BLIP-2 orchestrates a dynamic interplay between frozen image encoders, textual queries, and language descriptions. This vision-and-language representation learning stage establishes the foundation for effective cross-modal alignment. By harnessing the strengths of the frozen image encoders, this phase catalyzes the extraction of meaningful visual features, bridging the gap between visual and textual inputs.

The second stage of BLIP-2 is the visionary collaboration between the framework and frozen large language models (LLMs). This visionary alliance enables the framework to unlock the art of zero-shot instructed image-to-text generation. By invoking the dormant capabilities of LLMs, BLIP-2 transcends conventional boundaries and achieves the remarkable feat of generating textual output guided by natural language instructions.

The benefits of BLIP-2 are as follows. The far-reaching advantages of BLIP-2 confirm its groundbreaking status in the field of vision-language pre-training. First and foremost, BLIP-2 orchestrates a harmonious fusion of frozen pre-trained image models and LLMs. The innovative integration of a Querying Transformer (QFormer) in two distinct stages - representation learning and generative learning - propels BLIP-2 to the forefront of performance. In particular, BLIP-2 achieves state-of-the-art proficiency in various vision language tasks, including visual question answering, image captioning, and image-text retrieval.

Furthermore, BLIP-2 is a testament to the transformative power of LLMs, exemplified by OPT (Zhang et al., [17]) and FlanT5 (Chung et al., [18]). With the capacity for zero-shot image-to-text generation, BLIP-2 enters a new era of potential, encouraging capabilities such as visual knowledge reasoning and dynamic visual conversations.

BLIP-2's computational efficiency is enhanced by the strategic use of frozen unimodal models and a lightweight QFormer. This efficiency is clearly evident when compared to other state-of-the-art methods. In particular, BLIP-2 outperforms Flamingo (Alayrac et al., [19]) by a significant 8.7% on the zero-shot VQAv2 benchmark, while using only a fraction of the training parameters (54× fewer).

A final testament to BLIP-2's universality lies in its adaptability to advanced unimodal models. The extensive experimental results demonstrate BLIP-2's ability to effortlessly utilize superior unimodal models, confirming its status as a versatile tool for enhancing visual-language pretraining performance.

This project aims to advance VQA for educational videos by developing models based on online lectures and classes. We present a dataset compiled from machine learning lectures and open-ended questions and answers at Yeshiva University, Katz School of Science and Health [1]. To handle complex questions about course material, our model incorporates external knowledge resources, detailed modeling of visual and textual segments, and multistep reasoning, outperforming previous methods designed for other types of video. We believe our work will facilitate deeper video understanding and question answering capabilities for the educational domain.

3. Methods

3.1. Data Collection

Data collection for this study involves three distinct phases: image and transcription collection, and question and answer development. Each phase is critical to creating the data set necessary to analyze the machine learning course.

3.1.1 Image Collection

Starting with the image collection phase, this first step seems to be the least difficult. It consists of compiling slides from the PDF presentations of the machine learning course, which resulted in 885 images being collected. We used the PyMuPDF module [20], which converts the PDF slides to PNG images. This simple task lays the visual foundation for future analysis.

3.1.2 Transcript Collection

Transcript collection poses a significantly challenging task when moving forward. The primary aim of this phase is to transcribe the speech content from the recorded lectures on the machine learning course. The process starts by extracting audio from videos, which is necessary due to the restrictions of transcription tools that do not usually support video inputs. Subsequently, two distinct categories of transcription tools are used. The Python-based models include the Silero model [21, 22], the Wav2Vec Base model [23], Wav2Vec2 large-lv60 model [23], and the Google Speech-to-Text API [24]. Besides, professional online tools such as Cockatoo [25], Deepgram [26], Trint [27], Parrot [28], Veed [29], and Speechtext [30] are also used.

After an exhaustive test of ten different tools, the expected result is obtained: professional tools significantly outperform their Python model counterparts. The transcriptions generated by Wac2Vec and other Python models can only be described as catastrophic, making it impossible to form coherent sentences. This is not to say that professional tools always produce flawless results. Although many of their transcriptions produce accurate sentences, a significant number still contain errors, resulting in seemingly correct but inaccurate sentences. After a thorough comparison and evaluation, Deepgram [26] is identified as the preferred choice due to its remarkable accuracy.

Table 1: Speech-to-Text Tools Comparison

Tool Name	**Features and Capabilities**	**Online**
Silero Model [21]	Compact, supports multiple languages	
Wav2Vec Base Model [23]	Unsupervised learning, speech representations from raw audio	
Wav2Vec2 large-lv60 Model [23]	Improved architecture, large vocabulary support	
Google Speech-to-Text API [24]	Neural network models, supports 120+ languages	
Cockatoo [25]	Fast and accurate transcriptions, handles challenging conditions	✓
Deepgram [26]	Highly customizable, handles complex tasks	✓
Trint [27]	Supports multi-speaker and multi-language	✓
Parrot [28]	Fast and accurate transcription, streamlined process	✓
Veed [29]	Video editing tools, automatic subtitler	✓
Speechtext [30]	Speed and accuracy, handles audio and video files	✓

However, it is important to recognize that the low quality of the initial recordings is a significant challenge. The audio is riddled with noise, and even a small distance between the speaker and the microphone can make the content unintelligible. Therefore, although time-consuming, manual inspection of the content is a necessary part of this phase. The time required for manual inspection is significant. For example, for a two-hour video, manual inspection could take up to five times or more the length of the video. All the tools used in this project are listed in the Table 1.

3.1.3 Creating Questions and Answers

The final step is to create questions and answers for each slide. It is important to capture contextual clues that lead to the answers, along with metadata such as the week of the lecture and slide page. The curation of typically ten question-answer pairs for each slide involves consistently asking, ”What is the topic of this slide?” This critical inquiry aims to gather a comprehensive set of summary questions. For consistency and efficient retrieval, the sets of question-answer pairs are carefully organized and stored in a structured JSON format.

Table 2: QA Pair JSON Schema

Field	Description
Instruction	The question or instruction for the QA pair
Context	Contextual information, often including slide or transcript content
Response	The corresponding answer or response to the question
Category	The category of the QA pair, e.g., ’closed_qa’, ’information_extraction’
Week	The week of the ML course to which the content belongs
Page	The page number of the slide

Each JSON object is governed by the schema shown in the Table 2. Similar to the previous phases, this phase requires a significant amount of time due to its complexity.

3.2. Dataset Summary

In summary, the data collection journey involves three critical phases, each of which contributes significantly to the creation of a comprehensive data set. The meticulous curation of images, the intricate process of transcription, and the systematic generation of question-answer pairs collectively serve as the foundation for future analysis and learning.

Table 3: Breakdown of the collected data.

Data Type	Count
Question-Answer pairs	9,416
Transcripts	885
Images	885

The distribution of data types within this dataset is outlined in Table 3. The most significant component of the dataset is made up of 9,416 Question-Answer pairs. The dataset consists of 885 transcripts that provide detailed textual records of lecture content. Additionally, there are 885 images present in the dataset, each corresponding to a corresponding lecture slide. Each lecture slide has an associated transcript.

Table 4: Category Distribution

Category	Number
closed_qa	2,776
information_extraction	2,122
general_qa	1,125
open_qa	1,083
summarization	934
brainstorming	540
classification	510
creative_writing	326

3.3. The Structure of the Dataset

The dataset consists primarily of lecture slides in a visual format. Each slide is recorded as an image with a resolution of 960 by 540 pixels. As lecture slides, they carry information related to the specific topics of the lecture. These slides serve a dual purpose: they provide a visual summary and reference to textual data, while also representing the narrative of the lecture.

Data Type	Mean String Length	Max String Length
Answer	102.67	499
Question	45.85	122
Transcript	713.59	5,623

Table 5: Length Statistics for Instructions (Questions) and Responses (Answers).

Our textual dataset is categorized based on the nature of questions or tasks. As illustrated in the ”Category Distribution” Table 4, we have distinct categories such as closed-ended questions (closed_qa), information extraction, open-ended questions (open_qa), and more. The highest number of entries fall under the ”closed_qa” category, with 992 entries, while ”creative_writing” has the fewest, with 22 entries.

In addition, Table 5, titled ’Length Statistics for Instructions (Questions) and Responses (Answers)’, provides a comprehensive statistical analysis of the length of the questions, answers, and transcripts included in the dataset. On average, questions had a string length of 45.85 characters, answers had a string length of approximately 102.67 characters, and transcripts had an average length of 713.59 characters. The maximum string lengths recorded were 122 characters for questions, 499 characters for answers, and 5,623 characters for transcripts. These figures indicate a wide range of complexity and information density inherent in the data set.

3.4. Comparative Lexical Analysis of VQA Datasets

Our VQA dataset, characterized by an average question length of 45.85 words and an answer length of 102.67 words, offers an unprecedented level of lexical richness and complexity. This stands in stark contrast to the Visual Genome [31], where questions and answers average a mere 5.7 and 1.8 words, respectively. The substantial verbosity of our dataset not only offers a wider array of language usage but also introduces a challenging dimension for training VQA models, necessitating advanced language processing capabilities. In comparison to CLEVR [32], which presents questions averaging about 15 words, and GQA [33], with an average of 10 words per question, our dataset provides a more demanding linguistic landscape. The depth and breadth of linguistic expression in our questions and answers pose a beneficial challenge for developing robust models capable of nuanced comprehension and generation, significantly advancing the VQA domain.

3.5. Data Preprocessing

The data preprocessing is critical to optimize the formatting and preparation of raw data for subsequent training and modeling.

3.5.1 Merging and Structuring

Initially, datasets consisting of question-answer pairs, transcripts, and images were integrated based on shared attributes such as week and page numbers. This integration was crucial in obtaining a unified perspective of each data point.

Following the merge, the columns in the consolidated dataset were simplified and renamed to better reflect their content. For instance, columns previously labeled as 'instruction' and 'response' were accurately retitled 'question' and 'answer', correspondingly.

3.5.2 Dataset Splitting

Partitioning the data into training and validation sets is a critical step in the modeling process. For this project, we partitioned the data from weeks 1-11 and week 14 for training purposes, amounting to 8,681 samples, which is approximately 90% of the total data. This ensures a robust data set for the model to learn from. The remaining 10%, including weeks 15 and 16 with 735 samples, was reserved for validation to critically assess the model's performance on unseen data.

The precise numbers, divided into training and validation sets, are presented in Table 6.

Table 6: Distribution of samples across training and validation datasets.

Dataset	**Number of Samples**
Training	8681
Validation	735
Total	9,416

3.5.3 Dataset Creation

To streamline data retrieval and processing, we developed a customized dataset structure. The structure is illustrated below:

- Images corresponding to each question-answer pair were retrieved and loaded. The resolution of each image was resized to 224 by 224 pixels.
- A detailed textual prompt was created by combining the lecture transcript with the question and including a specific placeholder for the model to produce the answer.
- Due to the inherent limitations of the selected model, particularly when it comes to handling long sequences, any text that exceeds the token capacity of the model is shortened carefully.
- The correct answers were converted to a format which is suitable for the model and allows for both training and validation processes.

Separate datasets were curated for training and validation purposes, utilizing a customized dataset structure.

3.5.4 Data Loading

In order to retrieve data efficiently during training, a sophisticated data loading mechanism was employed. The objective of designing this mechanism was to:

- Sets of data points should be compiled accurately.
- Padding is used to standardize the length of sequences and ensure consistency across batches.
- Different types of data elements, such as images and textual cues, should be combined in a coherent structure that is suitable for modeling purposes.

Due to hardware limitations, the data was loaded in several batches, each containing a predetermined number of data points. The batching approach not only enhanced the efficiency of the training process but also uncovered the organization and dimensions of the training data, including images, textual prompts, and correct answers.

3.6. Model Architecture

3.6.1 LLaVA-1.5

The study employs the *LLaVA-1.5* model, a state-of-the-art multimodal architecture designed for general-purpose visual and language understanding. Developed by [34], *LLaVA* (Large Language and Vision Assistant, version 1.5) represents a significant advancement in combining a vision encoder and a large language model for multimodal tasks. Figure 2 depicts the block architecture of the model.

For our experiments, we used the *LLaVA-1.5* model. This model was designed to be efficient in training, achieving state-of-the-art performance on multiple benchmarks with just simple modifications and utilizing all public data. Its training completes in approximately 1 day on a single 8-A100 node. The model surpasses methods that use billion-scale data.

The *LLaVA-1.5* model was selected primarily for its robust and versatile architecture, which integrates different modules to handle various data types. Specifically, the model at its core includes:

1. **Vision Encoder**: Employing a pre-trained CLIP ViT-L/14 as the vision encoder, it translates visual content from images into a comprehensive set of visual features.

2. **Text Encoder**: Utilizing *LLaMA 2* as the text encoder, this component efficiently encodes textual data, capturing the nuances and intricacies of the language.

3. **Projection Matrix**: A simple projection matrix connects the vision encoder and the LLM, aligning the features from both modalities.

4. **Multimodal Fine-Tuning**: The model is fine-tuned for two different use scenarios: Visual Chat and Science QA, demonstrating its adaptability and effectiveness in various domains.

The strength of the *LLaVA-1.5* model lies in its ability to effectively integrate and understand content from both visual and textual modalities, producing precise and contextually relevant responses.

This modular and adaptive design of the *LLaVA-1.5* made it a suitable choice for our research, as it fit well with the multimodal nature of our dataset and the goals of our study.

3.6.2 BLIP

The Bootstrapped Language Image Pretraining (BLIP) model is an innovative approach in the field of AI and machine learning that focuses on multimodal learning, where both text and images are processed.

Multimodal Learning Framework: BLIP is designed for multimodal learning, i.e. it can understand and generate both textual and visual content. This is achieved by combining techniques from natural language processing (NLP) and computer vision [35].

Image encoder: The model uses the Vision Transformer (ViT) as an image encoder. This component is responsible for processing visual input and transforming it into a format that the model can understand and manipulate [35].

Text Encoder: Parallel to the image encoder, BLIP has a text encoder, often a variant of the Transformer model, which is used in NLP. This encoder processes textual input, allowing the model to understand and generate language [35].

Attention Mechanisms: A key feature of BLIP is its use of attention mechanisms, in particular cross-modal attention. This allows the model to focus on relevant parts of the text and image when generating outputs, leading to more accurate and contextually appropriate results [35].

Pre-training and fine-tuning: BLIP goes through a pre-training phase where it learns from a large dataset containing both text and images. This phase helps the model learn general representations of language and visual content. It can then be fine-tuned for specific tasks such as image captioning, text-to-image generation, or visual question answering [35].

Bootstrapping Technique: Unique to BLIP is its bootstrapping technique, in which the model uses its own predictions to further refine and improve its understanding of text-image relationships. This iterative process helps to improve the model's accuracy over time [35].

Applications: BLIP's architecture makes it suitable for a wide range of applications, including but not limited to automatic image captioning, visual question answering, and text-to-image generation. Its ability to understand and generate both text and images opens up possibilities for innovative applications in various domains [35].

3.6.3 Custom Model

As shown in Figure 1, the custom VQA model is a sophisticated neural network architecture designed for Visual Question Answering (VQA) tasks. It synergizes the capabilities of a Vision Transformer (DeiT) [36] for image feature extraction and a GPT-2 [37] encoder-decoder setup for text processing and generation. This fusion of visual and textual understanding enables the model to effectively interpret and answer complex questions about images.

3.6.4 Components

1. **Vision Transformer (DeiT)**: Extracts features from images and transforms them into a series of patches for detailed analysis.

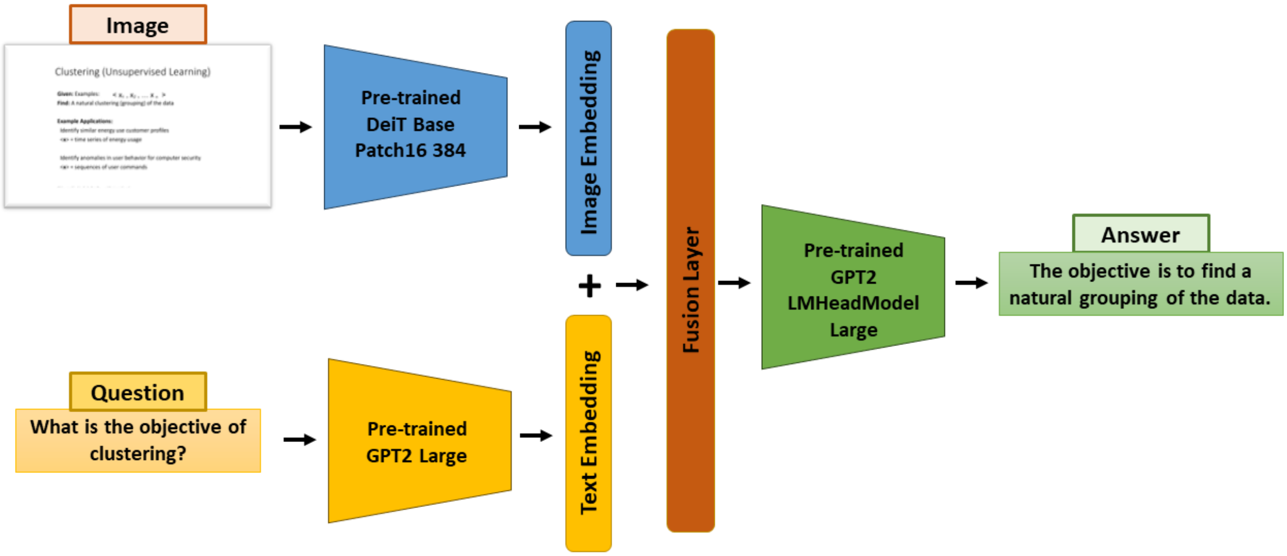


Image
Clustering (Unsupervised Learning)
Given: Examples: < x1 , x2 , … xn >
Find: A natural clustering (grouping) of the data
Example Applications:
Identify similar energy use customer profiles
<x> = time series of energy usage
Identify anomalies in user behavior for computer security
<x> = sequences of user commands
Pre-trained DeiT Base Patch16 384
Image Embedding
Question
What is the objective of clustering?
Pre-trained GPT2 Large
Text Embedding
+
Fusion Layer
Pre-trained GPT2 LMHeadModel Large
Answer
The objective is to find a natural grouping of the data.

Figure 1: The custom VQA model architecture includes a Vision Transformer (DeiT) for image encoding and a dual GPT-2 framework that acts as both an encoder and decoder for speech processing.

2. **GPT-2 Large. Encoder**: Processes text input into meaningful embeddings that capture context and semantics.

3. **GPT-2 Large with LM Head Decoder**: Generates contextually appropriate textual responses from the processed embeddings.

4. **Fusion Layer**: Merges visual and textual embeddings into a unified representation for effective response generation.

5. **Text Generation Process**: Uses iterative token prediction to generate coherent answers to visual questions using the combined embeddings.

3.6.5 Pix2Struct

Pix2Struct introduces an innovative framework in the field of visual language understanding [38]. It's designed to handle a wide variety of visual data, from web pages to mobile applications, by translating visual content into structured text. This model is distinguished by its unique approach to pre-training and its specialized architecture.

1. **Pretraining Strategy**: Pix2Struct's pre-training involves learning to parse masked screenshots of web pages into simplified HTML. This approach exploits the diverse visual elements of the web, effectively training the model to understand a variety of downstream tasks involving visually situated language.

2. **Architecture Overview**: Pix2Struct is based on the Vision Transformer (ViT) architecture, adapted to the specific challenges of visual language processing.

3. **Input Representation**: Unlike standard ViT models, Pix2Struct uses a variable resolution input representation. This means that input images are scaled to extract the maximum number of fixed-size patches to accommodate different resolutions and aspect ratios.

4. **Positional Embeddings**: To effectively handle variable resolutions, Pix2Struct uses 2-dimensional absolute positional embeddings for the input patches, improving the model's adaptability to different image sizes and formats.

5. **Robustness to Aspect Ratios**: The architectural changes in Pix2Struct provide robustness to extreme aspect ratios, which are common in various domains such as documents and mobile user interfaces.

6. **Adapting to Resolution Changes**: The model's input strategy allows for on-the-fly adjustments to sequence

length and resolution, making it highly versatile and suitable for various applications.

With this innovative architecture, Pix2Struct achieves state-of-the-art results on tasks in various domains, demonstrating its capabilities as a versatile and powerful tool in the field of document intelligence and visual language understanding.

3.6.6 BLIP-2-FLAN-T5-XLl

In the realm of advanced artificial intelligence, the BLIP-2-FLAN-T5-XL model represents a significant advancement in the synergy between vision and language tasks. Developed by Salesforce, this model integrates BLIP-2 (Bootstrapped Language Image Pretraining) for enhanced image-text recognition and interaction, the FLAN (Fine-tuned Language Net) methodology for adaptive language understanding, and employs the 'T5-XL' variant of the Text-to-Text Transfer Transformer architecture, denoting a large-scale, parameter-rich framework. The collective architecture and design of BLIP-2-FLAN-T5-XL aim to push forward the boundaries of multimodal artificial intelligence, enabling sophisticated understanding and generation of content across visual and textual domains.[39]

3.7. Training Hyperparameters

3.7.1 LLaVA-1.5

To train the LLaVA model [40] for multimodal understanding and visual chat applications, we meticulously selected a range of hyperparameters and utilized a custom dataset. This setup ensured optimal model performance, efficient convergence, and robust generalizability.

1. **Hardware Configuration**: Training was conducted on a device with 1 x A100 80GB GPU, supplemented by 8 vCPU and 62 GB RAM. This setup provided the computational power necessary for handling the large-scale multimodal data [34].

2. **Training Duration**: The model was trained for a duration of 11 hours, equivalent to 11 A100 GPU hours, balancing the computational intensity with the available resources.

3. **Batch Size**: We set the per device training and evaluation batch sizes to 4. This decision was based on the hardware's memory limitations and the model's computational requirements.

4. **Epochs**: Training was conducted for a total of 100 epoch, indicative of the large size of the training data and the efficiency of the learning algorithm.

5. **Learning Rate and Scheduler**: A learning rate of 2×10^{-4} was chosen with a cosine learning rate scheduler. The warmup ratio was set to 0.03, which facilitated a gradual increase in the learning rate during the initial phase of training, enhancing model stability [41].

6. **Gradient Accumulation**: We used gradient accumulation with a step size of 4. This approach effectively simulates the benefits of larger batch sizes, thus optimizing the training process within the constraints of available memory.

7. **Optimizer and Stability**: DeepSpeed with LoRA was used for optimization, enabling efficient training of large models with limited resources. The specific hyperparameters, including LoRA parameters (r = 128, alpha = 256) and mm_projector_lr (2e-5), were selected to achieve a balance between training efficiency and model performance.

8. **Fine-Tuning Strategy**: The model was fine-tuned on a custom dataset, formatted as a JSON file containing unique identifiers, image paths, and conversation data. This approach facilitated task-specific learning, enhancing the model's performance in generating tag-style captions for Stable Diffusion applications.

This training setup, leveraging the LLaVA model architecture, allowed for the development of a robust system capable of understanding and generating content from both textual and visual data, demonstrating state-of-the-art performance in multimodal tasks.

3.7.2 BLIP

1. **Hardware Configuration**: Training was conducted on a device with 1 x V100 16GB GPU, supplemented by 52 GB RAM. This setup provided the computational power necessary for handling the large-scale multimodal data.

2. **Training Duration**: The model was trained for a duration of 10 hours, equivalent to 10 V100 GPU hours, balancing the computational intensity with the available resources.

3. **Batch Size**: We set the per device training and evaluation batch sizes to 5. This decision was based on the hardware's memory limitations and the model's computational requirements.

4. **Epochs**: Training was conducted for a total of 70 epoch, indicative of the large size of the training data and the efficiency of the learning algorithm.

5. **Learning Rate and Scheduler**: A learning rate of 5×10^{-4} was chosen. [41].

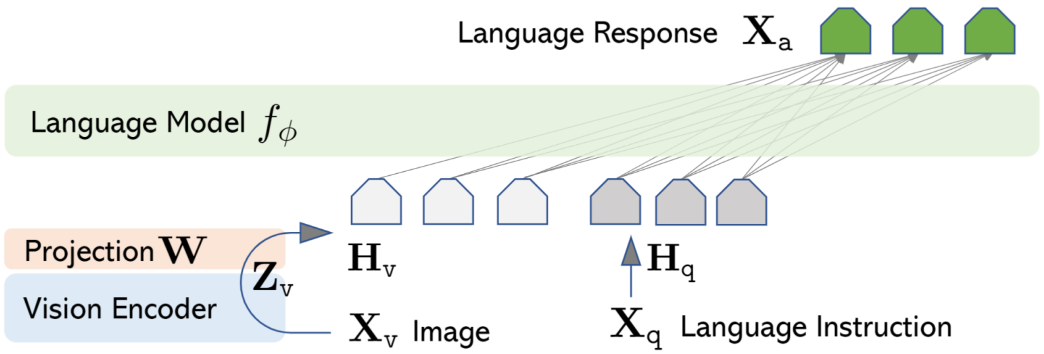


Figure 2: The pre-trained model architecture of LLaVA-1.5 includes a vision encoder and a large language model (LLM), as described in the reference [34].

3.7.3 Custom Model

The custom VQA model is an advanced neural network architecture for visual question answering, integrating a DeiT [42] for image feature extraction with a GPT-2 [43] encoder and decoder for text processing. This model employs a fusion layer to effectively combine visual and textual data, enabling the generation of accurate responses to complex queries.

1. **Hardware Configuration**: The training used a powerful computer with a 12th generation Intel i9-12900K CPU and 63.75 GB of RAM. It also had an NVIDIA GeForce RTX 3090 GPU for fast processing of large tasks and data.

2. **Training Duration**: The model underwent a rigorous training regimen for 3 days and 11 hours, a duration determined by computational constraints and the model's complexity.

3. **Batch Size**: Given the memory capacity of our hardware and the computational demands of the model, we opted for a training and evaluation batch size of 8 per device.

4. **Epochs**: The training process spanned 30 epochs, reflecting the extensive size of our dataset and the efficacy of our learning algorithm.

5. **Learning Rate and Scheduler**: We selected a learning rate of 1×10^{-6}, coupled with a ReduceLROnPlateau scheduler [44] (mode='min', factor = 0.1, patience = 5, verbose = True), to optimize learning efficiency.

6. **Optimizer and Stability**: The AdamW optimizer [45] (lr = 1×10^{-6}, betas = (0.9, 0.999)) was chosen to strike a balance between rapid convergence and model stability.

7. **Fine-Tuning Strategy**: Fine-tuning was performed on a custom dataset formatted as a JSON file containing unique identifiers, image paths, and conversational data. This tailor-made dataset enabled task-specific adaptation, thereby enhancing the model's ability to generate tag-style captions suitable for Stable Diffusion applications.

3.7.4 Pix2Struct

The Pix2Struct model [38] was strategically trained for advanced performance in document-based visual question answering. This section outlines the crucial hyperparameters and training configurations employed for optimizing this complex model.

1. **Hardware Configuration**: The training was done on a powerful computer with a 12th generation Intel i9 CPU and 63.75 GB of RAM. An NVIDIA GeForce RTX 3090 GPU was used for fast and efficient processing of large amounts of data.

2. **Training Duration**: Total training time spanned over 3 days, 16 hours, 32 minutes, and 13 seconds, reflecting the extensive computational efforts necessary for achieving high levels of accuracy and model refinement.

3. **Learning Rate and Optimizer**: An AdamW optimizer was employed with a learning rate of 1×10^{-5}. This choice of optimizer and learning rate was crucial for efficient and stable optimization of the model's parameters.

4. **Loss Function**: The training utilized the CrossEntropyLoss function, tailored to ignore padding tokens, thereby focusing the model's learning on meaningful and significant data points.

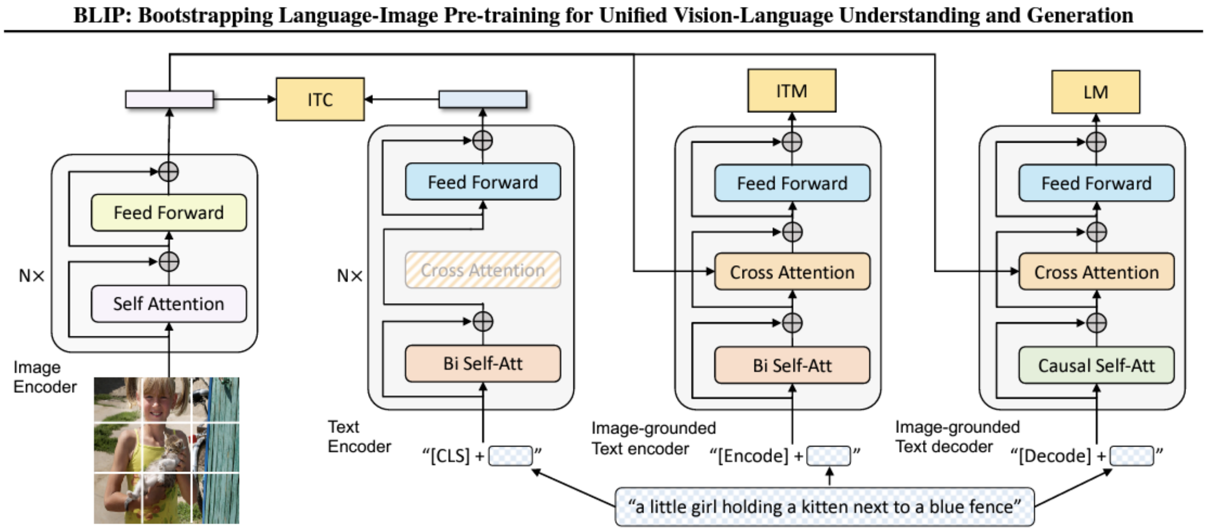


Figure 3: Architecture of the Bootstrapped Language Image Pretraining (BLIP) model [35]

5. **Gradient Accumulation and Epochs**: Pix2Struct's training involved gradient accumulation over 10 steps and was conducted across 5 epochs. This strategy balanced the need to process large-scale data while maintaining the model's training stability.

6. **Training Progression**: Throughout the training epochs, a consistent improvement in training and validation losses was observed, indicating enhanced model performance in terms of accuracy and predictive capabilities.

This comprehensive training setup for the Pix2Struct model underscores its potential as a state-of-the-art tool for document-based visual question answering, demonstrating its ability to handle and analyze complex multimodal data with high efficiency.

3.7.5 BLIP-2-FLAN-T5-XL

1. **Hardware Configuration**: Training was conducted on a device with 1 x A100 40GB GPU, supplemented by 83.5 GB RAM. This setup provided the computational power necessary for handling the large-scale multimodal data.

2. **Training Duration**: The model was trained for a duration of 20 hours, equivalent to 20 A100 GPU hours, balancing the computational intensity with the available resources.

3. **Batch Size**: We set the per device training and evaluation batch sizes to 5. This decision was based on the hardware's memory limitations and the model's computational requirements.

4. **Epochs**: Training was conducted for a total of 20 epoch, indicative of the large size of the training data and the efficiency of the learning algorithm.

5. **Learning Rate and Scheduler**: A learning rate of 5×10^{-5} was chosen.

4. Results

4.1. Training result

The comparative evaluation of the models, as summarized in Table 7, was conducted using three core metrics: ROUGE, COSINE, and BLEU. These metrics provided a comprehensive analysis across training and validation datasets, assessing n-gram overlap, semantic similarity, and translation accuracy.

For the ROUGE metric, the LLaVA-1.5 model outperformed others with a score of 41.10 in training and 35.50 in validation. The BLIP model scored 36.34 in training and 35.82 in validation, demonstrating consistency. The Custom Model, while lagging, showed a notable generalization ability with scores of 10.12 in training and 9.31 in validation. Pix2Struct followed closely with scores of 35.25 in training and 35.15 in validation. The BLIP-2-FLAN-T5-XL model, although lower, registered 12.83 in training and 12.038 in validation.

In terms of COSINE similarity, LLaVA-1.5 again led with 0.412 in training and 0.313 in validation. The BLIP model followed with a score of 0.373 in training and 0.374 in validation. The Custom Model scored 0.1392 in training and 0.1279 in validation, while Pix2Struct achieved impressive scores of 0.4002 in training and 0.3994 in validation. The BLIP-2-FLAN-T5-XL model scored 0.1830 in training and 0.1624 in validation.

For the BLEU metric, LLaVA-1.5 scored the highest with 0.513 in training and 0.497 in validation. BLIP scored 0.1638 in training and 0.1536 in validation, indicating potential for improvement. The Custom Model, with much lower scores of 0.0005 in training and 0.0004 in validation, needs significant refinement. Pix2Struct achieved scores of 0.0981 in training and 0.0889 in validation. The BLIP-2-FLAN-T5-XL model scored 0.0168 in training and 0.012 in validation.

Overall, these evaluations show the superior performance of the LLaVA-1.5 model across all metrics. The BLIP, Pix2Struct, and BLIP-2-FLAN-T5-XL models showed strong validation performance, with the Custom Model showing potential for future improvements, particularly in the transition from training to test environments.

Table 7: Evaluation metrics on training and validation sets for LLaVA-1.5, BLIP, Custom Model and Pix2Struct

Metric	**Training**	**Validation**
ROUGE		
LLaVA-1.5	41.10	35.50
BLIP	36.34	35.82
Custom Model	10.12	9.31
Pix2Struct	35.25	35.13
BLIP-2-FLAN-T5-XL	12.83	12.038
COSINE		
LLaVA-1.5	0.412	0.313
BLIP	0.373	0.374
Custom Model	0.1392	0.1279
Pix2Struct	0.4002	0.3994
BLIP-2-FLAN-T5-XL	0.1830	0.1624
BLEU		
LLaVA-1.5	0.513	0.497
BLIP	0.1638	0.1536
Custom Model	0.0005	0.0004
Pix2Struct	0.0981	0.0889
BLIP-2-FLAN-T5-XL	0.0168	0.012

5. Discussion

The results presented in table 7 highlight the comparative effectiveness of LLaVA-1.5, BLIP, Custom Model, Pix2Struct, and BLIP-2-FLAN-T5-XL in the VQA domain. The LLaVA-1.5 model's exceptional performance across all metrics underscores its ability to understand and generate speech, which is critical for VQA tasks. Its consistent performance from training to validation indicates a robust ability to generalize.

The BLIP model's stable ROUGE score and strong COSINE performance demonstrate its effectiveness in capturing semantic content. However, its BLEU scores suggest the need for improvements in linguistic translation accuracy. Similarly, the BLIP-2-FLAN-T5-XL model, while showing potential, indicates room for refinement, particularly in the area of semantic understanding, as reflected in its lower COSINE and BLEU scores.

Pix2Struct performs particularly well on the COSINE metric, highlighting its ability to understand and generate structured text from visually rich content. Its modest BLEU scores suggest a unique challenge in translation accuracy, a critical aspect of VQA that may require tailored approaches for visually situated language.

The Custom Model, despite lagging in overall performance, shows signs of effective generalization, which is promising for real-world applications. Its consistent results in training and validation suggest a solid underlying architecture that is ripe for targeted improvements.

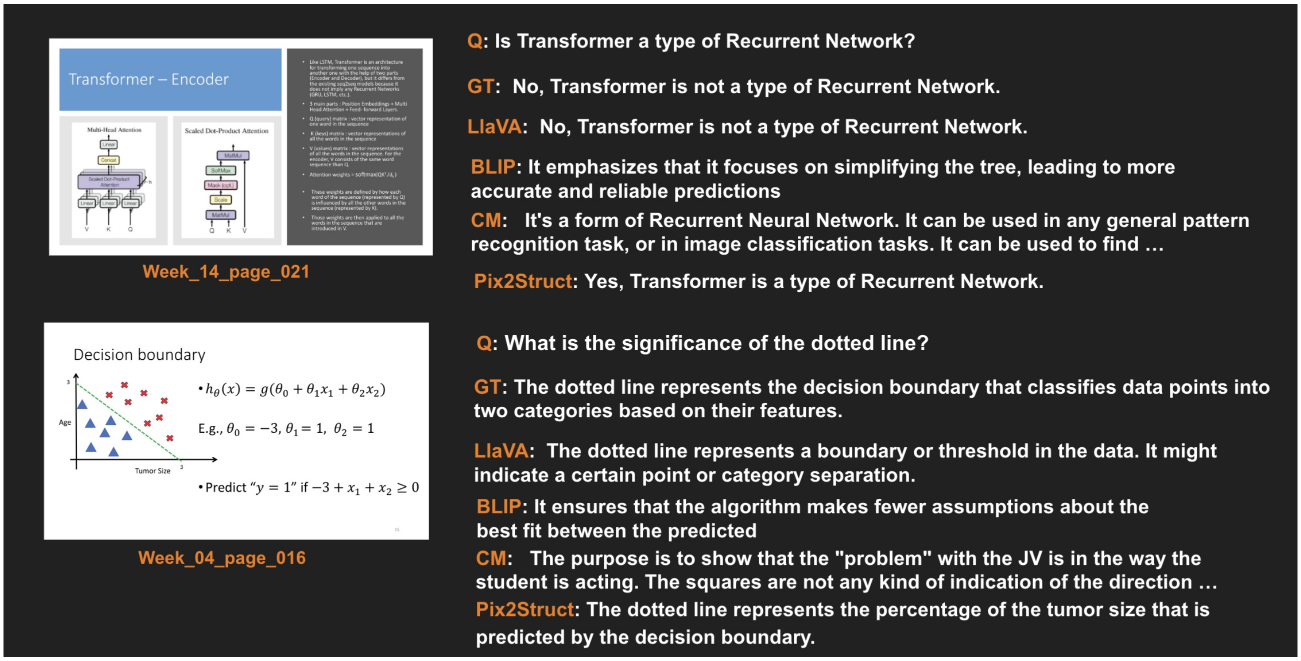
The varying performance of the ROUGE, COSINE, and BLEU metrics underscores the diverse nature of VQA tasks. The high ROUGE and COSINE scores are achieved by models such as LLaVA-1.5 and BLIP. However, the BLEU scores provide a contrasting view of translation accuracy, highlighting the complexity of achieving high performance in VQA systems.

In practice, the strengths of each model can be exploited in specific scenarios. The robustness of LLaVA-1.5 makes it suitable for a wide range of VQA tasks. Pix2Struct's unique capabilities for processing visually situated language could be ideal for tasks involving complex document structures. The BLIP-2-FLAN-T5-XL, with further refinement, could enhance its application in semantic understanding tasks.

In conclusion, this analysis underscores the diverse competencies required in VQA models and the need for a multifaceted approach in future research. In addition to improving quantitative metrics, a qualitative understanding of model behavior is essential for developing adaptable and robust VQA systems for real-world scenarios. The illustrative comparisons in Figures 4 and 5 provide additional insight into the real-world performance of each model, contributing to a deeper understanding of their strengths and areas for improvement.

6. Conclusion

This study conducted an in-depth comparative analysis of five models in the Visual Question Answering (VQA) domain: LLaVA-1.5, BLIP, Custom Model, Pix2Struct, and the newly included BLIP-2-FLAN-T5-XL. The analysis, using metrics such as ROUGE, COSINE, and BLEU, as detailed in Table 7, reveals distinct strengths and areas for



Transformer – Encoder

- Like LSTM, Transformer is an architecture for transforming one sequence into another one with the help of two parts (Encoder and Decoder), but it differs from the existing seq2seq models because it does not imply any Recurrent Networks (GRU, LSTM, etc.).
- 3 main parts : Position Embeddings + Multi Head Attention + Feed- forward Layers.
- Q (query) matrix : vector representation of one word in the sequence
- K (keys) matrix : vector representations of all the words in the sequence
- V (values) matrix : vector representations of all the words in the sequence. For the encoder, V consists of the same word sequence than Q.
- Attention weights = softmax(QK^T/d_k)
- These weights are defined by how each word of the sequence (represented by Q) is influenced by all the other words in the sequence (represented by K).
- Those weights are then applied to all the words in the sequence that are introduced in V.

Week_14_page_021

Q: Is Transformer a type of Recurrent Network?

GT: No, Transformer is not a type of Recurrent Network.

LlaVA: No, Transformer is not a type of Recurrent Network.

BLIP: It emphasizes that it focuses on simplifying the tree, leading to more accurate and reliable predictions

CM: It's a form of Recurrent Neural Network. It can be used in any general pattern recognition task, or in image classification tasks. It can be used to find …

Pix2Struct: Yes, Transformer is a type of Recurrent Network.

Decision boundary

- $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

E.g., $\theta_0 = -3, \theta_1 = 1, \theta_2 = 1$

- Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

Week_04_page_016

Q: What is the significance of the dotted line?

GT: The dotted line represents the decision boundary that classifies data points into two categories based on their features.

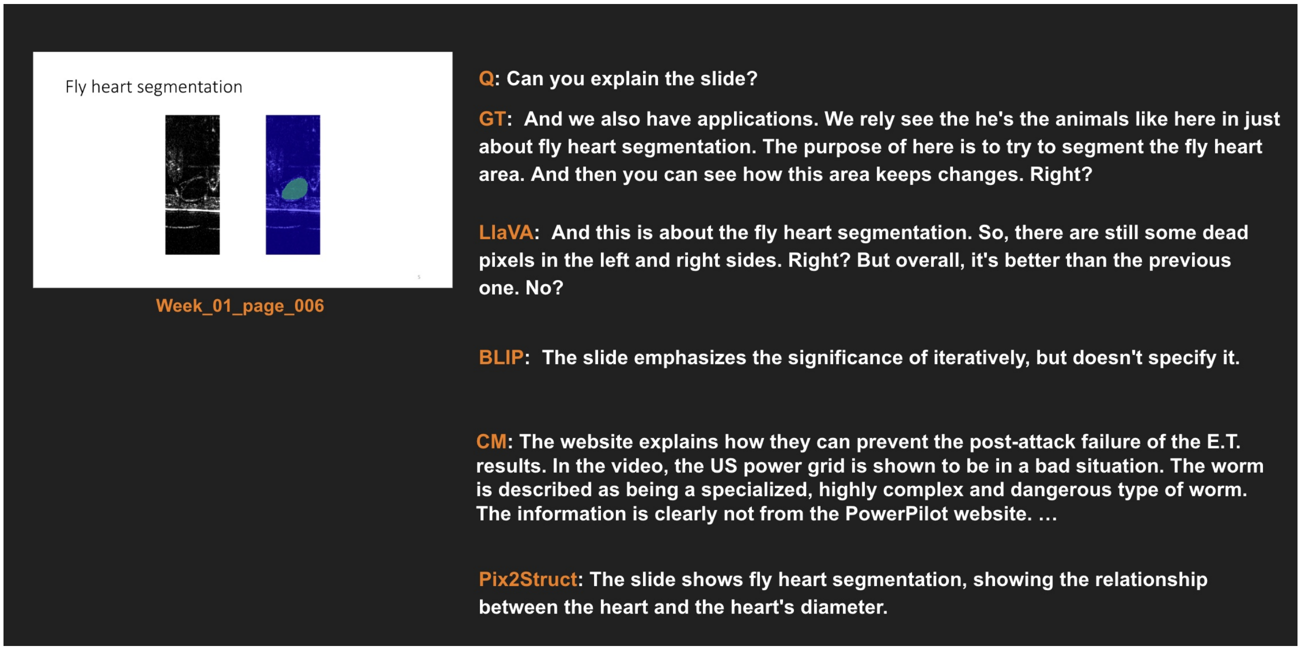
LlaVA: The dotted line represents a boundary or threshold in the data. It might indicate a certain point or category separation.

BLIP: It ensures that the algorithm makes fewer assumptions about the best fit between the predicted

CM: The purpose is to show that the "problem" with the JV is in the way the student is acting. The squares are not any kind of indication of the direction …

Pix2Struct: The dotted line represents the percentage of the tumor size that is predicted by the decision boundary.

Figure 4: Comparison of LlaVA vs. BLIP vs. Custom Model vs. Pix2Struct on random question



Fly heart segmentation

Week_01_page_006

Q: Can you explain the slide?

GT: And we also have applications. We rely see the he's the animals like here in just about fly heart segmentation. The purpose of here is to try to segment the fly heart area. And then you can see how this area keeps changes. Right?

LlaVA: And this is about the fly heart segmentation. So, there are still some dead pixels in the left and right sides. Right? But overall, it's better than the previous one. No?

BLIP: The slide emphasizes the significance of iteratively, but doesn't specify it.

CM: The website explains how they can prevent the post-attack failure of the E.T. results. In the video, the US power grid is shown to be in a bad situation. The worm is described as being a specialized, highly complex and dangerous type of worm. The information is clearly not from the PowerPilot website. …

Pix2Struct: The slide shows fly heart segmentation, showing the relationship between the heart and the heart's diameter.

Figure 5: Comparison of LlaVA vs. BLIP vs. Custom Model vs. Pix2Struct for the question ”Can you explain the slide?

improvement across these models. LLaVA-1.5 stood out for its superior performance, validating its effectiveness in high-accuracy VQA tasks. The BLIP model, while consistent, indicated areas for refinement in translation accuracy. Pix2Struct's strong performance in COSINE scores emphasized its capability with visually situated language, despite modest BLEU scores.

The BLIP-2-FLAN-T5-XL model, a new addition to this study, showed potential in certain aspects, but also highlighted the need for further development, especially in semantic understanding, as indicated by its COSINE and BLEU scores. The Custom Model, despite its current limitations, offers opportunities for potential improvement.

The results of this study underscore the complexity involved in developing VQA systems capable of both understanding and generating nuanced responses. While progress is evident, the results point to significant opportunities for improving translation accuracy and model adaptability across visual contexts.

Future research should focus on refining the linguistic capabilities of these models, with particular attention to improving BLEU score performance. The integration of multimodal data and external knowledge sources could also be beneficial in enriching the models' understanding and reasoning capabilities. In addition, the exploration of various architectural innovations, including advanced attention mechanisms and transformer-based approaches, could lead to further advances in VQA systems.

In conclusion, this research contributes important insights to the VQA community regarding model evaluation and development. It serves as a benchmark for future studies aimed at creating more sophisticated and context-aware VQA systems. The continued development of VQA models is critical to advancing the field of artificial intelligence by enabling machines to interact with visual data in a way that more closely resembles human cognitive processes.

References

[1] Yeshiva University, Katz School of Science and Health. https://www.yu.edu/katz/ai, 2023. Accessed: 2023-08-09. 1, 3

[2] Mateusz Malinowski and Mario Fritz. A multi-world approach to question answering about real-world scenes based on uncertain input. *CoRR*, abs/1410.0210, 2014. 1

[3] Yuke Zhu, Ce Zhang, Christopher Ré, and Li Fei-Fei. Building a large-scale multimodal knowledge base for visual question answering. *CoRR*, abs/1507.05670, 2015. 2

[4] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: Visual question answering, 2016. 2

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 2

[6] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and VQA. *CoRR*, abs/1707.07998, 2017. 2

[7] Weijie Su, Xizhou Zhu, Yue Cao, Bin Li, Lewei Lu, Furu Wei, and Jifeng Dai. VL-BERT: pre-training of generic visual-linguistic representations. *CoRR*, abs/1908.08530, 2019. 2

[8] Will Norcliffe-Brown, Efstathios Vafeias, and Sarah Parisot. Learning conditioned graph structures for interpretable visual question answering. *CoRR*, abs/1806.07243, 2018. 2

[9] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models, 2023. 2

[10] Makarand Tapaswi, Yukun Zhu, Rainer Stiefelhagen, Antonio Torralba, Raquel Urtasun, and Sanja Fidler. Movieqa: Understanding stories in movies through question-answering. *CoRR*, abs/1512.02902, 2015. 2

[11] Yunseok Jang, Yale Song, Youngjae Yu, Youngjin Kim, and Gunhee Kim. TGIF-QA: toward spatio-temporal reasoning in visual question answering. *CoRR*, abs/1704.04497, 2017. 2

[12] Jie Lei, Licheng Yu, Mohit Bansal, and Tamara L. Berg. TVQA: localized, compositional video question answering. *CoRR*, abs/1809.01696, 2018. 2

[13] Chenyou Fan. Egovqa - an egocentric video question answering benchmark dataset. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 4359–4366, 2019. 2

[14] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: collision events for video representation and reasoning. *CoRR*, abs/1910.01442, 2019. 2

[15] Yaoyao Zhong, Junbin Xiao, Wei Ji, Yicong Li, Weihong Deng, and Tat-Seng Chua. Video question answering: Datasets, algorithms and challenges, 2022. 2

[16] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models, 2023. 2

[17] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. Opt: Open pre-trained transformer language models, 2022. 2

[18] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models, 2022. 2

[19] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning, 2022. 2

[20] Pymupdf. https://pypi.org/project/PyMuPDF/, 2023. Accessed: 2023-12-04. 3

[21] Silero AI Team. Silero speech-to-text models. https://pytorch.org/hub/snakers4_silero-models_stt/, 2023. Accessed: 2023-06-30. 3

[22] Silero. Silero models: Pretrained enterprise-grade stt models. https://github.com/snakers4/silero-models, 2023. Accessed: 2023-06-30. 3

[23] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *CoRR*, abs/2006.11477, 2020. 3

[24] Google cloud speech-to-text api. https://cloud.google.com/speech-to-text, 2023. Accessed: 2023-06-30. 3

[25] Cockatoo: Online transcription service. https://www.cockatoo.com/, 2023. Accessed: 2023-06-30. 3

[26] Deepgram. Ai-powered speech recognition. https://www.deepgram.com/, 2023. Accessed: 2023-06-30. 3

[27] Trint. Automated transcription software. https://www.trint.com/, 2023. Accessed: 2023-06-30. 3

[28] Parrot: Online transcripts. https://www.parrot.us/, 2023. Accessed: 2023-06-30. 3

[29] Veed.io: Simple online video editing. https://www.veed.io/, 2023. Accessed: 2023-06-30. 3

[30] Speechtext: Fast and accurate audio transcription service. https://www.speechtext.ai/, 2023. Accessed: 2023-06-30. 3

[31] Ranjay Krishna, Yuke Zhu, Oliver Groth, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, 2017. 5

[32] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. *CoRR*, abs/1612.06890, 2016. 5

[33] Drew A. Hudson and Christopher D. Manning. GQA: a new dataset for compositional question answering over real-world images. *CoRR*, abs/1902.09506, 2019. 5

[34] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 6, 8, 9

[35] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022. 6, 10

[36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *CoRR*, abs/2012.12877, 2020. 6

[37] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. Preprint. 6

[38] Kenton Lee, Mandar Joshi, Iulia Turc, Hexiang Hu, Fangyu Liu, Julian Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. Pix2struct: Screenshot parsing as pretraining for visual language understanding, 2023. 7, 9

[39] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*, 2023. 8

[40] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 8

[41] Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with restarts. *CoRR*, abs/1608.03983, 2016. 8

[42] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 9

[43] Irene Solaiman, Miles Brundage, Jack Clark, Amanda Askell, Ariel Herbert-Voss, Jeff Wu, Alec Radford, and Jasmine Wang. Release strategies and the social impacts of language models. *CoRR*, abs/1908.09203, 2019. 9

[44] The reducelronplateau scheduler. https://pytorch.org/docs/stable/optim.html#torch.optim.lr_scheduler.ReduceLROnPlateau, 2023. Accessed: 2023-12-04. 9

[45] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. *CoRR*, abs/1711.05101, 2017. 9