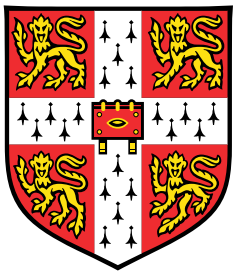
Incorporating Vision Encoders into Retrieval Augmented Visual Question Answering



Kristina Nikolić

Department of Engineering
University of Cambridge

This dissertation is submitted for the degree of
Master of Philosophy in Machine Learning and Machine Intelligence

Magdalene College August 2023

To my loving parents, who stirred my passion for learning and kept it shining brightly.
Your endless support has been a driving force behind my success.

To my dear brother Stefan, for all the stellar triumphs that await you.

Declaration

I, Kristina Nikolić of Magdalene College, being a candidate for the MPhil in Machine Learning and Machine Intelligence, hereby declare that this report and the work described in it are my own work, unaided except as may be specified below, and that the report does not contain material that has already been used to any substantial extent for a comparable purpose.

The code used in this work is written in Python. The software's starting point is the RA-VQA framework codebase (Lin and Byrne, 2022), and the MLMI-8 Practical 'In-Context Learning for Few-Shot Visual Question Answering' (codebase). The modifications required for the integration of a vision encoder, InstructBLIP Q-former (Dai et al., 2023), and detection of regions of interest are added. Code is made available here. The pre-trained transformer models are initialised using open-source Hugging Face transformers library (Wolf et al., 2020).

The word count, excluding declarations, bibliography, photographs and diagrams, but including tables, footnotes, figure captions and appendices, is 14,986 words.

Kristina Nikolić
August 2023

Acknowledgements

Firstly, I extend my deepest gratitude to my supervisor, Prof. Bill Byrne, for his gentle guidance and valuable suggestions throughout this project. His commitment to the MPhil students he supervises is truly commendable.

Additionally, I owe significant gratitude to my co-supervisor, Weizhe Lin. His patient mentorship, consistent availability, and dedication to ensuring I was always happy with my work have been invaluable.

Furthermore, I extend my appreciation to Xueyan Li for insightful discussions in the domain of visual question answering. It was a pleasure to work in a similar field.

I want to thank our Course Director, Professor John Dudley, for his unwavering kindness and understanding.

To my friends, Emilija Djordjević and Andrej Jakovljević, thank you for the many evenings spent together, immersed in our studies. Your company brought a special warmth to my Cambridge journey.

To Marko Macura, your unwavering love and steadfast presence have been my anchor through every challenge and triumph this year.

Lastly, I am forever grateful to my family and loved ones for their endless support.

Abstract

The Knoweldge-Based Visual Question Answering (KB-VQA) is a challenging task that requires image and natural language understanding together with access to external knowledge to answer the question regarding the image. The recent work often overlooks the importance of strong image understanding for this task. Commonly, the image is simply represented by textual descriptions, and the focus is shifted towards the improvement of knowledge retrieval methods. This thesis revisits the importance of image understanding for the KB-VQA task by proposing the use of the vision encoder to generate continuous image representations, arguing that this approach can result in a more comprehensive image representation.

We integrate an image encoder into the RA-VQA (Lin and Byrne, 2022), a baseline that originally relays on textual image descriptions. In order to improve the image understanding further, we propose the use of a question-aware mapping module to bridge the gap between the vision and language modalities by extracting the vision features most relevant to the question. Additionally, we do not just rely on global image representation, but we propose the procedure for extracting relevant image regions with respect to the question.

Our framework outperforms the RA-VQA baseline by large margin ($\sim 8.25\%$), achieving **62.56**% VQA score on OK-VQA benchmark (Marino et al., 2019). Our 4.5B parameters model outperforms many systems that use very large models, such as GPT-3 (175B), to obtain strong implicit knowledge retrieval and reasoning. Therefore, our results demonstrate the importance of powerful visual components for the KB-VQA system.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Large language models (LLMs) have witnessed fast-paced development in recent years (Brown et al., 2020; Driess et al., 2023; Touvron et al., 2023) due to extensive computation power and available data. This development of LLMs substantially caused increased interest in tasks that combine natural language understanding with other domains, such as vision. One such problem is Visual Question Answering.

Visual Question answering (VQA) is a task that lies in the intersection of scene understanding, natural language processing, and reasoning. The objective is to read an image and provide an answer to an accompanying question about the image content. The question and image pair from Figure 1.1 (a) are one example of a VQA task.

The VQA becomes more challenging when the image understanding is insufficient to answer the question. For example, questions such as "How many animals are in the image?" or "What is the colour of the shown dress?" can be answered with simple image reading, while the questions such as the one in Figure 1.1 (b) requires access to the information not directly available in the image. The VQA task that requires general knowledge which cannot be acquired from the image content we referred to as *Knowledge-based* VQA (KB-VQA).

The KB-VQA system takes three different information sources to predict the answer: input visual information (image), input question, and external knowledge. The current research in this domain mainly focuses on improving the incorporation of knowledge from external databases (Gao et al., 2022b; Gui et al., 2021; Lin et al., 2022; Marino et al., 2019), or finding the best techniques to utilise the knowledge of LLMs acquired during pre-training on large corpora (Hu et al., 2022b; Shao et al., 2023; Yang et al., 2022).

Meanwhile, the importance of the image understanding component of the KB-VQA systems is often overlooked. Recent work mainly relies on textual descriptions to represent an image. For example, RA-VQA (Lin and Byrne, 2022) uses a set of off-the-shelf models to generate textual features such as image captions and object descriptions. Although suitable

for later direct prompting of LLMs, these types of generic textual representations cannot comprehensively describe the image and, therefore, can miss the detailed visual information needed for answering the question (see Figure 1.2) (Hu et al., 2022b; Lin et al., 2022).

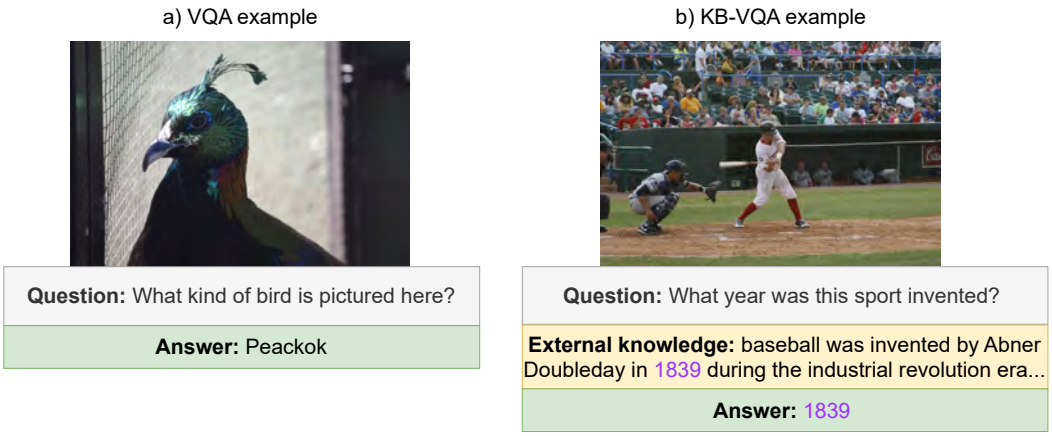


Fig. 1.1 **The example of (a) VQA and (b) KB-VQA image-question pair**. The KB-VQA task requires information not contained in the image to answer the question.

In our work, we focus on improving the image understanding of the KB-VQA system. We argue that continuous image representation, obtained by encoding the image with a vision encoder, can overcome the weaknesses of textual image descriptions, leading to improved overall image understanding. To test our hypothesis, we incorporated an image encoder into the RA-VQA framework, which originally utilized only textual descriptions as image representation.

Additionally, we explore how to better exploit visual representations generated by a frozen vision encoder. Firstly, we propose the use of *question-aware* mapping module (Q-former) (Dai et al., 2023) designed to bridge the gap between the vision and language modalities. Instead of generating static image representations, the Q-former extracts visual features tailored to a specific image-question pair.

Furthermore, inspired by Lin et al. (2022), we do not just rely on global image representation, but we also extract *regional* image features. We introduce a procedure that detects relevant image regions based on the question. These detected regions are then processed by the vision encoder to obtain continuous regional image representations.

1.1 Contributions

The main contributions of this thesis are summarised here:

- We propose a question-aware Q-former and frozen vision encoder to extract visual features tailored for the given KB-VQA question. This approach results in more comprehensive image representation, surpassing a text-based vision approach and significantly boosting accuracy.
- We propose regional feature extraction by selecting the image regions of interest based on the question. Our results suggest that the region-based approach outperforms the whole image-based and sliding window-based approaches.
- We show that vision representations obtained with a vision encoder and simple MLP mapping network can complement text-based vision, improving overall performance. Furthermore, we demonstrate the importance of pre-training a mapping module that is used to bridge the gap between vision and language modalities.
- Our framework outperforms RA-VQA baseline (Lin and Byrne, 2022) by large margin ($\sim$ 8.25%). It achieves **62.56**% VQA score surpassing the SOTA models within the same parameter scale. Our 4.5B parameters model outperforms many systems that use very large models, such as GPT-3 (175B), to obtain strong implicit knowledge retrieval and reasoning. Therefore, the performance of our framework demonstrates the importance of powerful visual components for the KB-VQA system.

Fig. 1.2 **The VQA example illustrating the loss of information after image-to-text transformation**. The caption used to represent an image does not provide the necessary information to answer the question, even though the information is present in the image.

1.2 Thesis Overview

The structure of the thesis is as follows:

In Chapter 2, we introduce the concept of Vision-Language Models (VLMs) and we discuss two approaches to visual feature extraction, clarifying our preference towards the image encoding approach. We also describe the architecture of Q-former, an instruction-aware module we will use to bridge the modality gap between the vision encoder and language model.

In Chapter 3, we direct our attention to the domain of Knowledge-Based Visual Question Answering. We present a unified-view of the relevant studies and then delve into a detailed description of our baseline, the RA-VQA framework.

In Chapter 4, we present our approach to enhancing vision understanding of RA-VQA. We propose our framework, RA-VQA-Vis, that incorporates a vision encoder and mapping module to generate continuous image representation. We propose two architectures for the mapping module: a simple multi-layer-perceptron pre-trained on captioning task, and a more complex, transformer-like, question-aware Q-former. We conclude by defining the procedure for selecting regional image representations with respect to the question asked.

In Chapter 5, we provide the experimental setup. We define the concrete model versions used for each framework component, describe the relevant datasets, and define the VQA evaluation metric.

In Chapter 6, we present our results. We start by positioning our best model with respect to the SOTA systems and discussing whether the continuous vision features complement text-based image descriptions. Then we systematically test the contribution of each system component, and conclude by presenting the results of MLP mapping network pre-training.

In Chapter 7 we summarise the thesis and suggest the direction of future work.

Chapter 2

Background

In this chapter, we introduce relevant topics necessary for the understanding of our work. We start by introducing the concept of Vision-Language Models (VLMs) in Section 2.1, where we define two learning strategies commonly used for these systems. Next, we delve deeper into the visual components of the VLMs, discussing the two approaches to visual feature extraction in Section 2.2. Here, we also clarify our preference towards the *image encoding* approach. Finally, we provide a detailed explanation of the architecture of Q-former, an instruction-aware mapping module we will use to bridge the gap between vision and language modalities (Section 2.3). Collectively, this provides a foundation for the review of the relevant work in the domain of Knowledge-Based Visual Question Answering covered in Chapter 3.

2.1 Vision-Language Models

Vision-Language Models (VLMs) represent a set of models which aim to jointly process visual and natural language data. The development of VLMs has been influenced by the widespread adoption of Transformer architectures (Vaswani et al., 2017). Transformers are designed to model long-range dependencies better than RNN-based approaches while increasing models' throughput, enabling training on significantly larger data sets. As a result, the design of VLMs has shifted from hand-crafted image descriptions and pre-trained word vectors, towards the use of transformer-like image and text encoders.

Leaning strategy. Recent VLMs consist of three main components: image encoder, text encoder, and training strategy, to jointly or separately learn the representations of image and text. There are several actively used learning strategies showing good performance on downstream tasks (Dosovitskiy et al., 2020; Gan et al., 2022; Li et al., 2023; Radford et al., 2021; Wang et al., 2021). Here, we describe two of them relevant to our framework:

- **Contrastive objective:** Aligning images and texts to a joint feature space in a contrastive manner.
- **Vision-conditioned text-generation:** Insert visual information into the space of the language model to retain its generative power.

2.1.1 VLMs with Contrastive Objective

The contrastive loss aims to encourage mapping paired data points to the vectors close to each other in the joint space while minimising the distance between unpaired ones. The VLMs following this objective (Alayrac et al., 2022; Jia et al., 2021; Li et al., 2021; Radford et al., 2021) commonly use separate text and vision encoders for mapping vision and language inputs into the joint embedding space. These dual encoders, trained on large vision-language datasets, can produce highly generic textual and visual representations suitable for various downstream tasks. In our work, we leverage the vision encoder from CLIP (Radford et al., 2021) to extract vision futures from images, later used for the KB-VQA task.

2.1.2 VLMs for Vision-Conditioned Text-Generation

For the VQA task, in addition to having a good image understanding, a model needs to have strong reasoning and text-generation abilities to answer the given question. The contrastive dual encoder approach described in the previous section can extract powerful image representations, however, it lacks good generative abilities (Radford et al., 2021). Therefore, for question answering, we want to use different group of VLMs specifically designed to have strong generative power.

Vision-conditioned LM. This group of VLMs aims to employ text-generation abilities of pre-trained large language models (LLMs). Instead of learning image and text representations with two separate encoders, these models attempt to *insert visual information into the space of pre-trained LLMs* thus (1) preserving LLM text-generation abilities and (2) utilizing knowledge of LLM gained with pre-training (Alayrac et al., 2022; Chen et al., 2023b; Dai et al., 2023; Driess et al., 2023; Li et al., 2023; Mokady et al., 2021; Wang et al., 2021).

Encoder-decoder structure. For fussing the visual information into the LM, we will follow the approach of SimVLM (Wang et al., 2021) and VL-T5 (Cho et al., 2021) who uses encoder-decoder architecture and pass both the visual information and input text (e.g. VQA question) to the LM encoder. The LM encoder then forms joint vision-text encoding that is later used by the decoder for the response generation (see Figure 2.1).

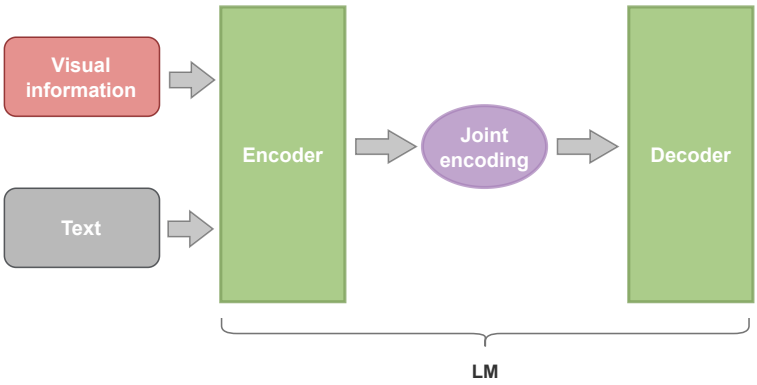


Fig. 2.1 **The illustration of VLM with vision-conditioned text generation and encoder-decoder structure**. Visual information and input text are passed to the LM encoder which generates joint encoding feed to the decoder.

2.2 Visual Features Generation

Having outlined the high-level VLM architecture for the VQA task, we will now detail how we can extract information from an image, later integrated into the LM encoder (Figure 2.1).

There are two main approaches in the literature, and they both make use of pre-trained models for visual feature extraction. We described both approaches below, highlighting their benefit and limitations.

2.2.1 Image-to-Text

A popular choice for visual feature extraction is the use of specialised models to extract textual image descriptions. For example Gao et al. (2022a); Gui et al. (2021); Lin and Byrne (2022), use image caption, OCR, and detected objects to describe an image. Such set of textual descriptions we denote as *text-based vision*.

Image-to-text transformation is an attractive approach because generated textual features can directly be fed to the LM together with the additional text, using the LM's embedder (see Figure 2.2 a)). However, there are two inherited limitations of this approach:

- **Loss of relevant information.** Textual description of an image can encapsulate limited information, which is often very general (e.g. image caption, detected objects, etc.) and therefore, may not contain necessary knowledge for completing the task. For example, let us assume that the image is described with the caption "A group of people dancing in the park". If we want to use this image description for the task of VQA, to answer the question "How many people are in the park?", we would experience a loss of information - even though it may be present in the image, the information on a

number of people is not preserved in the caption. In these situations, the model can only guess the correct response based on the limited information acquired.

- **Multiple specialised models required.** In order to generate a textual description of an image, numerous models have to be used. For example, captioning models (Li et al., 2020c; Zhang et al., 2021), object detection models (Ren et al., 2015; Zhang et al., 2021), and OCR models (Du et al., 2020). This introduces additional architectural and computational complexity to already complex multi-model systems for vision-language understanding.

In our work, we aim to overcome these limitations with the use of pre-trained image encoders, to extract encoded image features without transformation to text form. This approach is described in the next section.

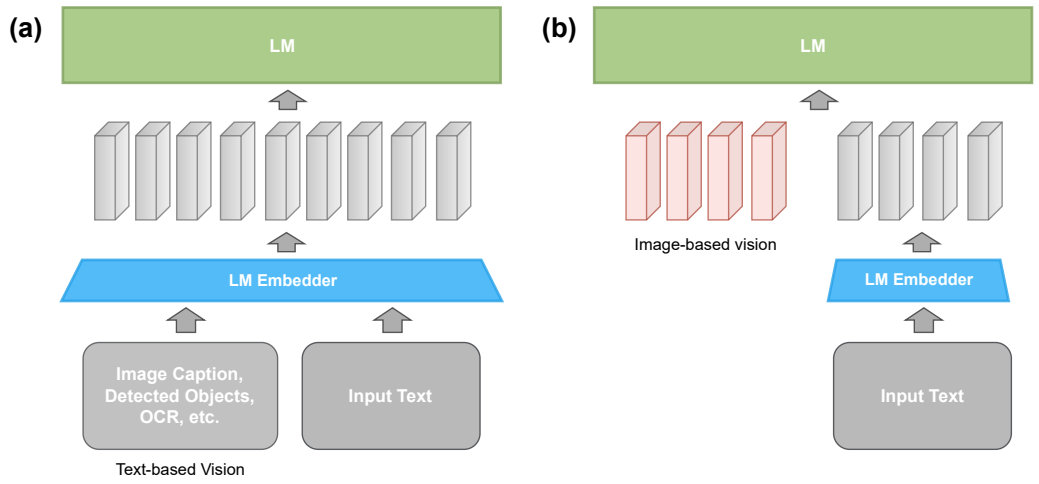


Fig. 2.2 **LM conditioned on text-based vision features (a), and on image-based vision features (b).** Textual image descriptions are encoded using LM's embedder, while the image-based vision embeddings have to be generated separately.

2.2.2 Image Encoding

Alternative ways of generating vision features uses pre-trained vision encoders to learn continuous image representations. Recently developed vision encoders (Dosovitskiy et al., 2020; Jia et al., 2021; Li et al., 2021; Radford et al., 2021) posses rich generic representations of images in the form of image embeddings that can be passed to the LM without previous transformation to the text form. These vision-encoders are often trained in the contrastive

manner as described in Section 2.1.1, and are therefore suitable for future alignment with the embedding space of LM encoders.

Alignment with LM. The challenge of this approach is aligning the embedding space of the vision encoder to the space of the language model. Unlike in the previous case, we cannot use LM's embedder to generate embeddings understood by the LM encoder (see Figure 2.2). Instead, we use the additional mapping model to map the output of the vision encoder into the *embedding space of the language model*. Usually, this mapping module has simple architecture such as a fully connected linear layer (Eichenberg et al., 2021; Lin et al., 2022; Tsimpoukelli et al., 2021), however recent work (Li et al., 2023) introduces a transformer-like mapping module specifically designed to bridge the gap between vision and language encoders - Q-former (Section 2.3).

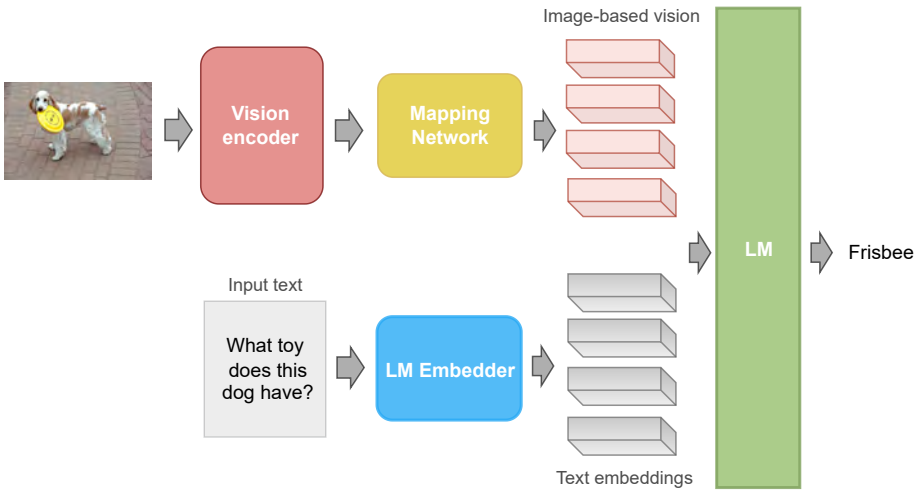


Fig. 2.3 **VLM diagram with vision-conditioned text generation using vision encoder and mapping network.** The diagram illustrates the VQA task. Image is encoded with vision encoder, whose output is mapped to the space of LM using mapping network to form *image-based vision*. The input text (VQA question) is embedded with LM Embedder and fed to the LM together with image-based vision.

The whole system diagram of VLM with vision-conditioned text generation is illustrated on Figure 2.3. Visual features obtained in the described way we denote as *image-based vision*. We argue that image-based vision may overcome the information loss limitation described in Section 2.2.1, and hence provide a more comprehensive image representation than text-based vision. In our experiments, we will test this hypothesis on the KB-VQA task.

2.3 Q-former - Instruction Aware Mapping Module

In the previous section, we described a VLM system based on vision-conditioned text generation. As we discussed, an important segment of this system is the mapping module, the component used to map image embeddings into the space of the language model, providing alignment of vision and language modalities. In this section, we will dive deeper into the architecture of one concrete mapping module - Q-former.

In the previous work (Eichenberg et al., 2021; Lin et al., 2022; Tsimpoukelli et al., 2021), the mapping modules typically had relatively simple architectures, with the most popular choice of a fully-connected linear layer. However, the recent research on bridging the gap between vision and text encoders, BLIP2 (Li et al., 2023), and Instruct-BLIP (Dai et al., 2023), introduce a more complex, transformer-based mapping module for this purpose. Namely, BLIP2 propose Q-former, a lightweight querying transformer to connect frozen image encoder and frozen LLM.

2.3.1 Q-former's Architecture

Q-former is designed to extract informative visual features from the frozen image encoder and to align them with the embedding space of the LM. As shown in Figure 2.4, Q-former consists of two transformer sub-modules that share self-attention layers: an image transformer and a text transformer. The image transformer uses a set of learnable query embeddings to extract a fixed number of output vision features from the vision encoder. The queries interact with each other through self-attention layers and with the frozen image features through cross-attention layers. The text transformer can function as both a text encoder and a text decoder. The learnable queries interact with the text through the same self-attention. This architecture results in 188M parameters.

2.3.2 Pre-training Procedure

The authors carefully designed a two-stage pre-training procedure to encourage the Q-former to extract relevant vision features that can be aligned with the language models. The pre-training process is described below.

1. Representation learning stage. In the first stage, the Q-former is only connected to the frozen vision encoder and trained using image-text pairs. Join-training with three pre-training objectives and an appropriate self-attention mask is performed. The objectives used are:

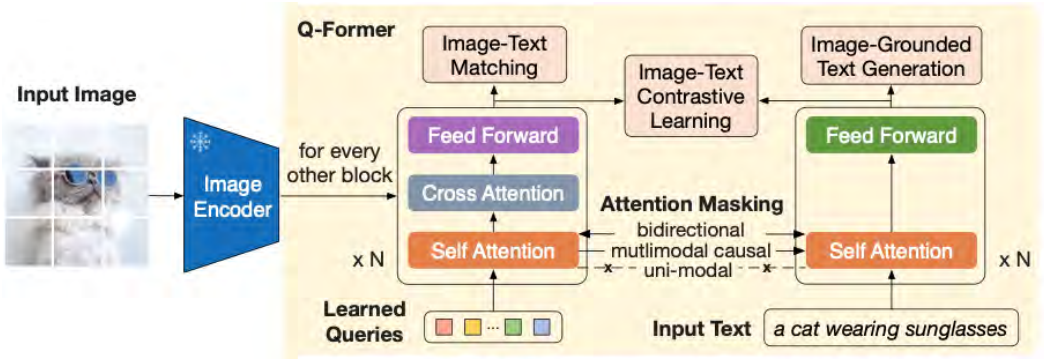


Fig. 2.4 **Q-former's architecture and pre-training objectives.** The lightweight querying transformer consists of vision and text transformers which share self-attention layers. The set of learnable queries extracts a fixed number of vision features from the frozen image encoder. The queries interact with the two transformers through self-attention and with vision features through cross-attention. Figure source: Li et al. (2023).

i **Image-text contrastive learning.** Model learns to align image and text representations obtaining maximal mutual information.

ii **Image-grounded text generation.** The vision transformer module learns to extract comprehensive vision representations passed to the text transfer module for text generation.

iii **Image-text matching.** The model aims to predict whether the image-text pair is matched, ultimately learning the detailed correlation between image and text representations.

2. Generative pre-training phase. In the second pre-training phase, the frozen LM is added to the system. It is connected with the Q-former via a fully-connected linear layer that matches the query output dimensions with the LM hidden size. Therefore, the Q-former now functions as an information bottleneck between the frozen vision encoder and the frozen LM.

2.3.3 Instruction Aware Visual Feature Extraction

Having introduced the Q-former's design, we'll now highlight one of its key advantages that influenced our decision to integrate the Q-former module into our system. Below, we describe the instruction-aware visual feature extraction ability of the Q-former.

So far, we referred to the vision feature extraction as a task-agnostic process. Namely, the vision encoder produces a static image representation regardless of the subsequent task.

However, if we anticipate varying task instructions for the same input image, generating vision representations tailored to the specific task would be undeniably beneficial.

Even though the Q-former's text transformer module can be conditioned on the instruction by design, BLIP2 opts for an instruction-agnostic approach. It leaves the Q-former's input text field vacant, possibly not maximizing the Q-former's capabilities.

InstructBLIP Q-former. The InstructBLIP proposes instruction-aware Q-former. Extending BLIP2 approach, the authors condition the Q-former on the instruction string, stimulating it to extract vision features most relevant for the concrete task (Figure 2.5). Inspired by its performance, we incorporated the InstructBLIP Q-former into our framework. Namely, for the task of VQA, the instruction string can be replaced by the given question. In Section 4.4, we further develop this idea.

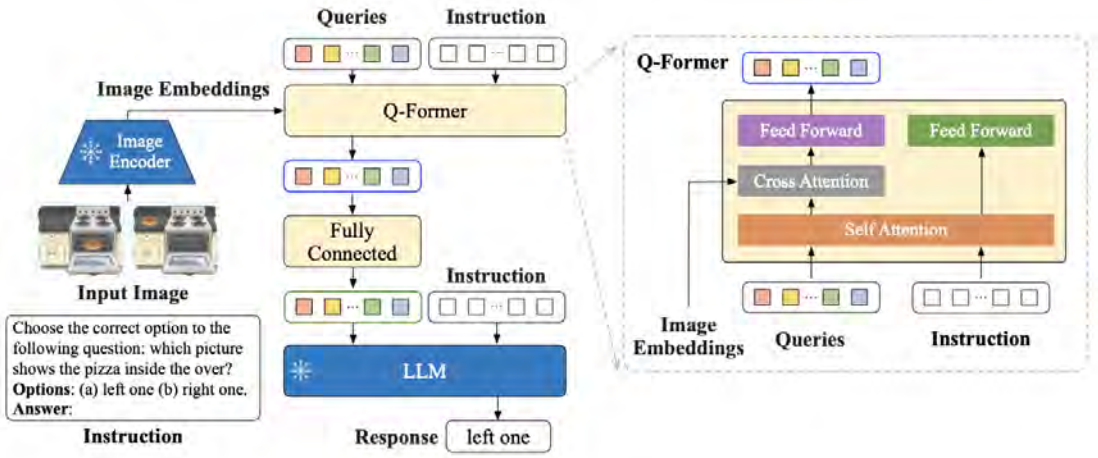


Fig. 2.5 **InstructBLIP model architecture.** The set of learnable query embeddings extracts instruction-aware visual features from the frozen vision encoder. The Q-former's output is propagated to the frozen LLM through the fully connected layer. Figure source: Dai et al. (2023).

2.4 Conclusion

In this chapter, we have provided the necessary context upon which the proceeding chapters rely. In Section 2.1, we introduced the concept of VLMs and defined the learning strategies our system will rely on. In Section 2.2, we delineated two approaches to visual feature extraction and laid out arguments for our preference: the image encoding approach. Finally, we provide a detailed explanation of the Q-former architecture and its pre-training procedure

(Section 2.3). Here, we highlight the Q-former's ability to extract visual features conditioned on the task definition, which will be an important aspect of our work. (Section 2.3).

In the following chapter, we turn our attention to Knowledge-Based Visual Question Answering. We present a unified view of the state-of-the-art approaches to this problem, and we introduce the RA-VQA framework as the main baseline for this thesis.

Chapter 3

Knowledge-Based Visual Question Answering

Having laid down the fundamental concepts for understanding vision-language systems in the previous chapter, this chapter now turns its attention directly to the domain of Visual Question Answering. Firstly, we provide a unified view of the state-of-the-art research in Section 3.1. Following that, we delve into a detailed description of the RA-VQA framework, which stands as the main baseline for this thesis (Section 3.2).

3.1 Relevant Work

The KB-VQA frameworks can be roughly separated into two segments: the VLM modules for vision-language understanding and the optional component for knowledge retrieval. In this section, we categorize relevant systems based on: (1) their VLM's method for image understanding; (2) their technique for knowledge retrieval. We will begin by discussing relevant work in these categories, and then we will provide a unified overview of the state-of-the-art models in Section 3.1.3.

3.1.1 Vision-Language Systems

Early work. The core of VQA task lies in joint vision and language understanding. Early studies in this domain can be roughly grouped into three categories with respect to multi-modal modelling: (1) Visual and textual features can be combined via cross-modality fusion (Guo et al., 2021; Jiang et al., 2020; Singh et al., 2019; Yu et al., 2019, 2018); (2) Multi-modal transformers can be trained from scratch on large scale image-text pairs, and then fine-tuned for VQA task (Li et al., 2020a, 2019; Lu et al., 2019; Su et al., 2019; Tan and Bansal, 2019);

(3) and visual and language representations can be aligned by contrastive learning (CLIP (Radford et al., 2021), ALIGN (Jia et al., 2021)). Although these models achieve notable performance on the visual language tasks such as image-text retrieval, they lack the strong text-generation ability and reasoning necessary for the KB-VQA task (Radford et al., 2021).

LLMs for Vision-to-Language Generation. Inspired by the rapid advancements in LLMs, recent VQA system designs now revolve around leveraging pre-trained LLMs for enhanced language understanding, reasoning, and answer generation. Given that VQA is a *vision*-language task, modern VQA frameworks employ two approaches for fusing visual representations into the LLM. They either perform image-to-text transformations to acquire image descriptions that can be directly fed into the LLM (see Section 2.2.1), or they generate continuous vision representations, which are then injected into the LLM's embedding space (see Section 2.2.2).

Textual Image Representation

One approach to allowing LMs to understand images is projecting images into textual features with the use of pre-trained vision models such as captioning models (e.g. Oscar+ (Li et al., 2020c)) or object detectors (e.g. VinVL (Zhang et al., 2021)). After image-to-text transformation, the VQA is treated as a pure NLP task. Recent KB-VQA frameworks: TRiG (Gao et al., 2022b), RA-VQA (Lin and Byrne, 2022), KAT (Gui et al., 2021), and PICa (Yang et al., 2022) showed that this approach is feasible. Furthermore, Prophet (Shao et al., 2023) prompt GPT-3 (Brown et al., 2020) with textual answer heuristics to generate an answer, achieving notable KB-VQA performance on benchmarks such as OK-VQA (Marino et al., 2019), while PromptCap (Hu et al., 2022b) argues that generic image captions cannot comprehensively describe images, and it proposes a question-aware captioning model for relevant caption generation.

Continuous Image Representation

Although representing images in a text form may be convenient for their further use in LLMs (especially if the model is only accessible via API), converting finite text descriptions risks excluding detailed visual information needed to answer the question (Hu et al., 2022b; Lin et al., 2022). Therefore, in the last year, the research focus shifted towards using powerful vision encoders to generate continuous image representations. The REVIVE (Lin et al., 2022) is the first to revisit the importance of vision understanding for answer generation. It employs a ViT-based (Dosovitskiy et al., 2020) vision transformer to generate continuous feature representations of both the whole image, and relevant image regions.

Furthermore, PaLI (Chen et al., 2023b), and PaLI-X (Chen et al., 2023a) highlight the gap in power distribution of vision and language components of modern VLMs and by scaling up the vision encoder, achieve remarkable performance on OK-VQA benchmark, outperforming models that rely on large LLMs such as GPT-3, and explicit knowledge retrieval. The current SOTA model, PaLM-E (Driess et al., 2023), scales the vision and language model further, presenting the 562B parameters VLM (25B for vision encoder), which makes it the current largest VLM.

All of the above methods employ simple mapping for aligning vision and language representations. However, BLIP2 (Li et al., 2023) propose Q-former, a specialised transformer-like mapping module to bridge the modality gap between the vision encoder and language model (see Section 2.3). InstructBLIP (Dai et al., 2023) enhance this module by using it for instruction-aware visual feature extractions (see Section 2.3.3).

3.1.2 Knowledge Retrieval

The KB-VQA task is challenging because the answer cannot be obtained only by image understanding; it requires additional domain knowledge or commonsense reasoning. Recent work can be grouped with respect to the type of knowledge they rely on. While early work is more focused on retrieving knowledge from external knowledge bases (KB) (*explicit knowledge*), the recent models rely on the exceptional capabilities of LLMs to employ the knowledge obtained during the pre-training on large corpora (*implicit knowledge*). In this section, we discuss relevant approaches to knowledge retrieval by grouping systems into these two categories.

Explicit Knowledge Retrieval

As a natural approach, early studies developed KB-VQA systems that can access external knowledge bases for *explicit knowledge retrieval*. The external knowledge can be in both structured forms, such as ConceptNet and other knowledge graphs (Li et al., 2020b; Marino et al., 2021; Wu et al., 2022), or in unstructured forms, such as Wikipedia passages (Gao et al., 2022b; Gui et al., 2021; Lin et al., 2022). Most of these knowledge retriever models are based on the dual encoder framework proposed in the DPR (Karpukhin et al., 2020) or its variations (Gui et al., 2021; Lin and Byrne, 2022; Lin et al., 2022). The DPR encodes both query and knowledge entities with pre-trained BERT encoders (Devlin et al., 2018) and calculates their similarity score by taking the dot product of two encoded dense vectors.

Similar to the image understanding discussion, the KB-VQA frameworks can be unified with respect to the type of image features they use for explicit knowledge retrieval:

(1) **Text-based image representation.** TRiG and RA-VQA generate text-based features, such as image captions, OCR and object tags (detected object classes), and combine them with the question to form a query for DPR-based knowledge retrieval. The retrieved knowledge is later fed to the answer generator model (LM). Additionally, RA-VQA proposes joint training of retrieval and the answer generator to encourage selecting knowledge entities that provide meaningful information for answer generation.

(2) **Continuous image representation** On the other side, KAT and REVIVE use continuous image representations generated with a vision encoder to query the KB. They employ contrastive trained vision and language encoders from CLIP to compute the alignment of the query and the KB entity. The REVIVE extends KAT by introducing object-centric image representations.

Implicit Knowledge Retrieval

Motivated by the promising capacities of LLMs (e.g. GPT-3), another line of research relies on LLMs as the source of *implicit knowledge*, therefore, either fully disregarding explicit knowledge retrieval (PICa, PromptCap, Prohpet), or combining both explicit and implicit knowledge resources (KAT, REVIVE). The recently developed group of models utilise GPT-3 to generate an answer directly. PICa uses the off-the-shelf captioning model to prompt the GPT-3 with the image caption, questions and a few in-context examples. Following PICa, PromptCap develops a question-aware captioning model to generate image descriptions dependent on the asked question. Finally, Prophet uses a knowledge-free VQA model to generate answer candidates and relevant few-shot examples and combine them to prompt the GPT-3, achieving notable results on the OK-VQA benchmark.

On the other hand, KAT and REVIVE use separate answer generator that combines explicit knowledge retrieved based on continuous image representations and implicit knowledge retrieved by prompting GPT-3 with textual image description.

The general purpose LLMs, such as InstructBLIP, PaLI, and PaLM-E, when tested on KB-VQA task, make use of their LLMs as only knowledge sources which, combined with the strong vision components of these VLMs, allows them to achieve SOTA performance on OK-VQA benchmark.

3.1.3 Unified View of SOTA Models

In the previous section, we discussed recent work in KB-VQA. In this section, we present a unified view of SOTA models[1], grouping them with respect to their strategy for knowledge retrieval (Figure 3.1) and with respect to the features they use for image understanding (Figure 3.2). We choose these levels of abstraction because we recognise them as the major methodological differences among recent work. Table 3.1 provides more detailed information on the type of vision features used.

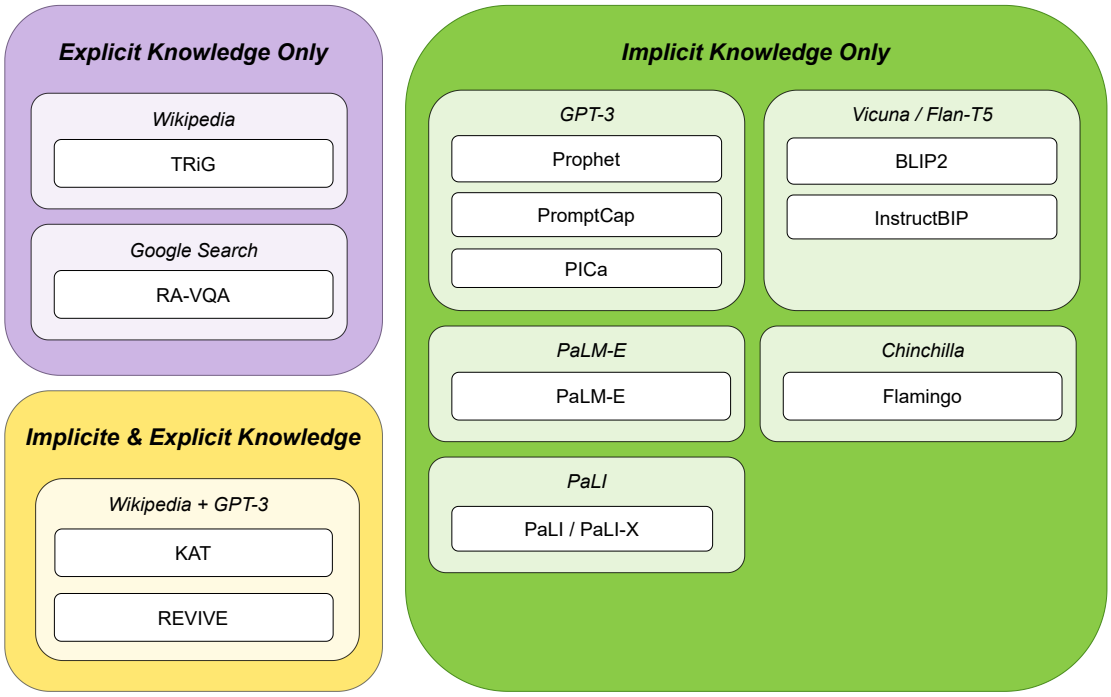


Fig. 3.1 **The subset of KB-VQA systems grouped by the knowledge retrieval strategy.** Many models rely on GPT-3 and other LLMs for implicate knowledge retrieval. The TRiG and RA-VQA rely on external knowledge bases. KAT and REVIVE unify both implicate knowledge from GPT-3 and explicit knowledge.

[1]Models are chosen from the leader-board https://paperswithcode.com/sota/visual-question-answering-on-ok-vqa (last modified on May 29, 2023). We also include BLIP-2 and InstructBLIP, which reported their performance on OK-VQA on June 15, 2023.

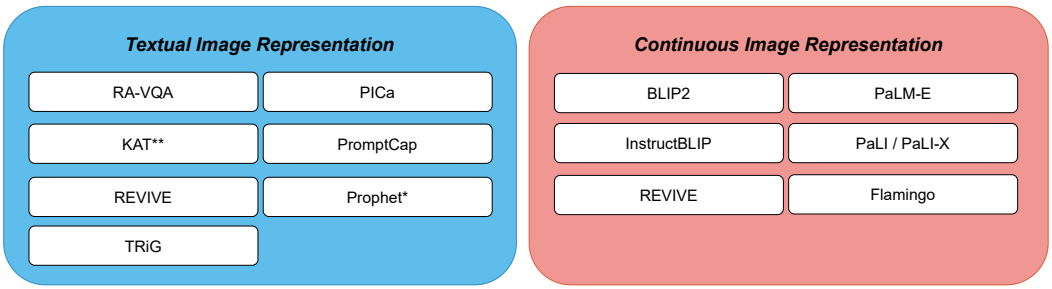


Fig. 3.2 **The subset of KB-VQA systems grouped based on the type of vision representation used for answer generation.** Textual representations are image descriptions such as captions and detected object tags. The continuous representation is obtained by encoding the image with a vision encoder. KAT** uses continuous image representation only for knowledge retrieval. Prophet* uses continuous representation for generating textual heuristic fed to the LM.

Table 3.1 **The overview of SOTA KB-VQA systems.** Models are grouped based on their strategy for knowledge retrieval. We provide the type of visual features fed to the answer generator and the knowledge source (LLM for implicate knowledge and KB for explicit knowledge). OC stands for object-centric continuous representation. Flamingo* is not fine-tuned for the KB-VQA task. The RA-VQA is the baseline for this thesis.

Model	Image Representation		Knowledge Source
	Textual	Continuous	
Explicit Knowledge			
RA-VQA	Captions + Objects + OCR	-	Google Search
TRiG	Captions + Objects + OCR	-	Wikipedia
Implicit Knowledge			
PaLM-E	-	✓ + OC	PaLM-E
PaLI-X	-	✓	PaLI-X
PaLI	-	✓	PaLI
InstructBLIP	-	✓	Vicuna / Flan-T5$_{XXL}$
BLIP-2	-	✓	Vicuna / Flan-T5$_{XXL}$
PICa	Captions	-	GPT-3
Prophet	Captions	-	GPT-3
PromptCap	Captions	-	GPT-3
Flamingo*	-	✓	Chinchilla
Explicit & Implicit Knowledge			
REVIVE	Captions + Objects	✓ + OC	Wikipedia + GPT-3
KAT	Captions + Objects	-	Wikipedia + GPT-3

After providing an overview of the relevant studies, we will now delve deeper into the specifics of the RA-VQA system, which stands as the primary baseline for this thesis.

3.2 Retrieval Augmented Visual Question Answering

Retrieval Augmented Visual Question Answering with Outside Knowledge (RA-VQA) (Lin and Byrne, 2022) is the multi-modal OK-VQA framework and main baseline for this thesis. RA-VQA introduces joint training of document retriever and answer generator, outperforming recent KB-VQA systems that perform explicit knowledge retrieval. The framework can be divided into three main segments: Image Understanding (on which we focus), Documents Retrieval, and Answer Generator. The modules are described in detail below.

3.2.1 Image Understanding

The authors of RA-VQA opt for the image-to-text approach for representing an image (introduced in Section 2.2.1). The set of pre-trained vision models is utilised to:

1. Detected objects and their attributes
2. Generate image caption
3. Detect string within the image (OCR)

The details of each step are given below.

Object detection

Pre-trained object detection model VinVL (Zhang et al., 2021) is used to extract objects and their attributes. For example, brown white standing dog, brown floppy long ear, brown white head, etc. for Figure 1.2.

Image Captioning

The pre-trained captioning model Oscar+ (Zhang et al., 2021) is used for caption generation. One sentence-long image caption provides the main observation from an image and should encapsulate interactions between the visual elements. For example: "A men sitting *on* a bench *in* a flooded park".

OCR

The KB-VQA questions can require understanding the text strings present in the image. To be able to answer this type of question, the Google OCR (Optical Character Recognition) APIs are used to extract the text strings in the images.

Next step. After the textual image descriptions are generated, they are combined with the question and passed to the next pipeline segment, the Documents Retriever.

3.2.2 Document Retriever

RA-VQA relies on leveraging an outside knowledge base (KB) (e.g. Wikipedia) to retrieve information needed to answer the question. The document retrieval method proposed by RA-VQA is based on Dense Passage Retrieval (DPR) (Karpukhin et al., 2020).

DPR consists of two transformer-like encoders, document encoder E_d and query encoder E_q. The goal is to retrieve documents that contain information relevant to the query. For the concrete KB-VQA example, this relevance is computed as follows. Firstly, the question is concatenated with the text features extracted in the previous step (image caption, objects with attributes and OCR) to form the input string x. Then, x is encoded using query encoder $E_q(x) \in \mathbb{R}^h$, while documents are encoded using document encoder $E_d(d) \in \mathbb{R}^h$. Finally, the relevance of document d for the input string x is defined by the inner product between the obtained representations:

$$r(x,d) = E_q^T(x)E_d(d) \tag{3.1}$$

The RA-VQA document retriever aims to maximise this score $r(x,d)$ when the document is helpful for answering the question.

After computing the relevance score for all the documents in the KB, the DPR score $p_\theta(.|x)$ is calculated for the K most relevant documents:

$$p_\theta(d_k|x) = \frac{exp(r(x,d_k))}{\sum_{j=1}^{K} exp(r(x,d_j))} \tag{3.2}$$

DPR score represents retrieval's confidence in each of K selected documents, which plays a role in the answer selection process.

3.2.3 Answer Generation

After obtaining text-based features and relevant documents, the generative encoder-decoder model, with parameters ϕ, is fed with the question and text features (x) and the selected document (d_k) to generate the answer:

$$y_k = \arg\max_y p_\phi(y|x, d_k) \tag{3.3}$$

Each of the K documents is considered, and the final answer is selected as the one with the highest joint probability:

$$\hat{y}, \hat{d} = \arg\max_{y, d_k} p_\phi(y|x, d_k) p_\theta(d_k|x) \tag{3.4}$$

3.2.4 Joint Training of Document Retriever and Answer Generator

In the recent KB-VQA approaches (Gao et al., 2022c; Gui et al., 2021), the document retriever is often kept frozen. However, RA-VQA proposes joint training of these two modules to encourage retrieval of the documents that actually led to the generation of the correct answers.

To formulate the joint training loss $\mathscr{L}_{RA-VQA}$, the retrieved documents are divided into two groups: $P^+(x,S)$ and $P^-(x,S)$, where x is a string used to retrieve documents, and S is the set of answer annotations. The group P^+ contains documents which 1) contain the answer from S (by string match) and 2) are used when the model generated the most popular answer from S[2] (Equation 3.3). The group P^- contains documents with no answer from S, and which did not cause the model to generate the most popular answer (i.e., are not helpful).

Finally, the training of an RA-VQA system is achieved with the loss

$$\mathscr{L}_{RA-VQA} = -\sum_{(x,S)\in T} \Big(\sum_{k=1}^{K} \log p_\phi(s_k^*|x, d_k) + \sum_{k\in P^+(x,S)} \log p_\theta(d_k|x) - \sum_{k\in P^-(x,S)} \log p_\theta(d_k|x)\Big). \tag{3.5}$$

The first term of the $\mathscr{L}_{RA-VQA}$ aims to improve answer generation based on the question, extracted text-based features and retrieved document. The subsequent terms are designed to influence the document retrieval aspect: the second term promotes the retrieval of useful documents (P^+), while the third term aims to reduce the ranking scores of the ones considered unhelpful (P^-). Therefore, both document retrieval and answer generation are improved during training.

[2]Each question has 10 available answers from annotators. The most popular one is the one with the most votes.

3.3 Conclusion

We began this chapter by providing a comprehensive overview of the relevant work in the KB-VQA domain in Section 3.1, and we followed with the introduction of the RA-VQA framework and a detailed definition of its key components in Section 3.2.

This sets the scene for our next chapter, where we highlight the weaknesses of the RA-VQA image understanding component and propose our modification for improved vision understanding: the integration of a vision encoder for image-based feature generation.

Chapter 4

Approach

We begin this chapter by highlighting the weaknesses of RA-VQA, a framework introduced in the previous chapter. Then, we propose modifications to enhance its image understanding component. We describe our approach in Section 4.2, defining our framework architecture followed by a formal definition of each component.

Later on, we turn our attention to two distinct variants for the mapping module component. In Section 4.3, we detail our approach for the pre-training of the MLP module. Section 4.4 delves into our strategy for integrating the question-aware Q-former module into RA-VQA. In Section 4.5, we include regional image representation by proposing the procedure for detecting regions of interest based on the question.

We conclude the chapter with Section 4.6, defining the prompt format employed in our work.

4.1 Weaknesses of Text-Based Vision for VQA

Now that we described our baseline, the RA-VQA framework, we can highlight what we see as weaknesses in its image understanding module and propose a solution.

As disused in Section 3.1, most recent SOTA methods, including RA-VQA, use only text-based features combined with the question and retrieved knowledge as input to the final answering model. Therefore, to some extent, they treat the VQA problem as a pure natural language processing (NLP) task by providing an answer generator with only textual descriptions of the image.

We highlight two main weaknesses of this approach which both emphasise an inevitable loss of information during the image-to-text transformation:

1. **Insufficient descriptive features.** Transforming image embeddings from a vision encoder (such as VinVL (Zhang et al., 2021)) to the text form (list of detected objects, object attributes, image caption, etc.) can lead to extensive loss of relevant information required to answer the question. For example, consider an image of the dog with the caption "`A dog sitting on the couch`". This caption does not provide enough information to answer the question such as "`What breed of dog is this?`". The information on the dog breed may be present in the image but disregarded to make the caption concise. Although this problem is intuitive (it is not realistic to comprehensively describe an image with a short caption), previous work often relies on such insufficient, descriptive textual features.

2. **Relationship and interactions among objects.** After object detection is performed, the list of objects and their attributes is fed to the model. These lists do not preserve information on the interactions between the objects. The captioning is often used with ideas to compensate for this by including object interaction such as: "`A man is carrying a child.`" However, as described above, one sentence long caption may fail to focus on the information needed to answer the question. The relationship among objects and their relative positioning is important and should not be neglected but carefully extracted from the image.

Proposed solution - Image-Based Vision. We argue that the problems described above can be mitigated with the direct pass of vision-encoded image embeddings to the answer generator without previous transformation to the text form. We propose encoding both the whole image and the object-centric region of interest to reduce the loss of information during the image-to-text transformation.

4.2 RA-VQA-Vis Architecture

Our architecture design is built on top of the RA-VQA framework (Lin and Byrne, 2022). As an improvement, we add system components required for generating a visual representation of an image, as introduced in 2.2.2. Concretely, we incorporate vision encoder and mapping network into the RA-VQA framework, introducing its extension, RA-VQA-Vis framework, where "Vis" stands for **Vis**ion encoder.

In this section, firstly, we give the system overview, pointing out the modifications proposed by us, and then we describe individual system components in more detail.

4.2.1 System overview

The system overview is given in Figure 4.1. Motivated by Dai et al. (2023); Li et al. (2023); Lin et al. (2022); Zhu et al. (2023), we introduce a frozen vision encoder to extract image representations. These image representations are further aligned with the LM embedding space using a mapping network and then passed to the answer generator model. The answer generator model (encoder-decoder LM) is conditioned on image-based vision, retrieved knowledge, and text-based vision. Text-based vision contains an image caption, detected objects with attributes, and text strings detected within the image. The document retriever model retrieves passages from the knowledge database conditioned on text-based vision and the VQA question.

To highlight our modification to the RA-VQA framework, we compare the original RA-VQA and our system (RA-VQA-Vis) in Figure 4.2. The introduced vision encoder is kept *frozen* while the mapping network is *trained*.

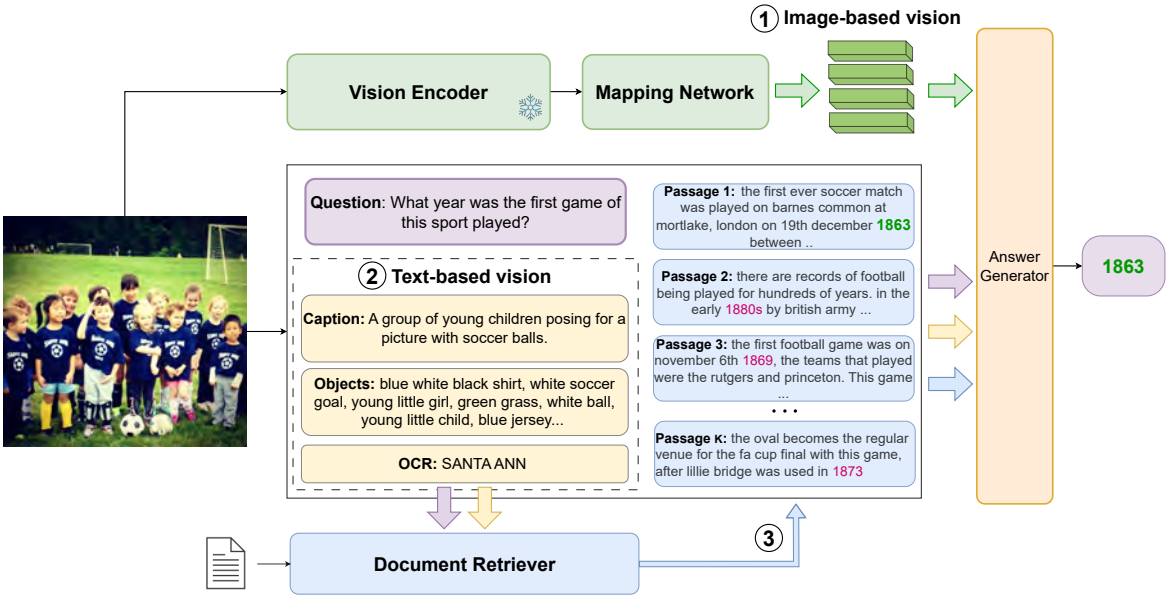


Fig. 4.1 **The RA-VQA-Vis system diagram.** The introduced vision encoder and mapping network (green) generate continuous image representations (image-based vision). The answer generator is conditioned on (1) image-based vision, (2) text-based vision (image caption, OCR, and detected objects), (3) retrieved documents, and the question. The document retriever is conditioned on text-based vision and the question.

Now that we have seen the system overview, we can formally define each of its components in the following sections.

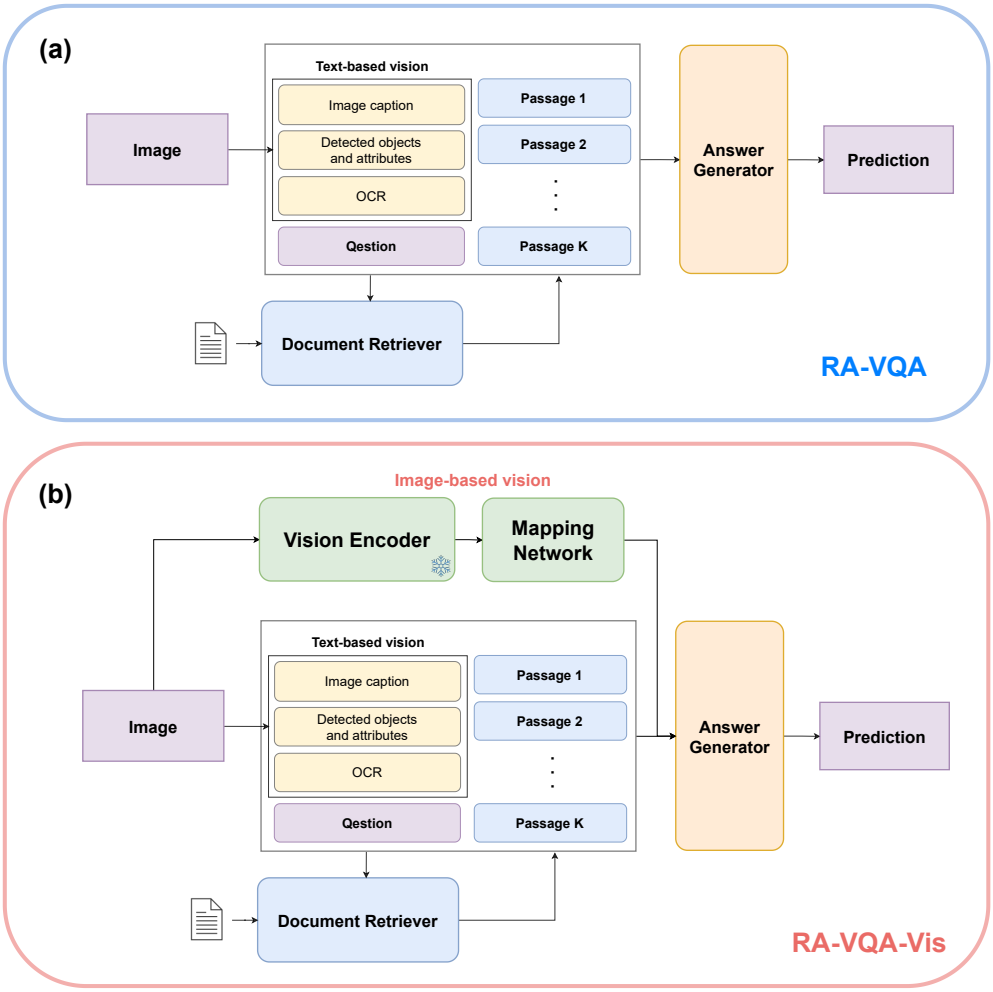


Fig. 4.2 **Side-by-side comparison of (a) RA-VQA and (b) RA-VQA-Vis.** Extending the RA-VQA design, and with the goal of improved image understanding, we introduce the vision encoder and mapping module employed for image-based feature generation.

4.2.2 Vision Encoder

The input image is firstly passed to the vision encoder $v(.)$. The vision encoder maps an input image I into the continuous vector representation $V = v(I)$, where $V \in \mathbb{R}^r$, and r is the image embedding dimension.

Essential pre-training. We use an encoder that is pre-trained with the VLM contrastive objective introduced in Section 2.1.1. We see this type of pre-training as an essential starting point for the future alignment of the image embeddings into the embedding space of the answer generator. Following Dai et al. (2023); Li et al. (2023); Lin et al. (2022); Zhu et al. (2023), we use an already powerful vision encoder (such as CLIP's (Radford et al., 2021) or ALIGN's (Jia et al., 2021)) and keep it *frozen*.

Our main contribution to the RA-VQA framework architecture is the introduction of this vision encoder component together with mapping module described in the next section.

4.2.3 Mapping Module

The vision encoder output is mapped to the LM's embedding space using the learnt mapping network $m(.)$. Concretely, the r-dimensional image embedding V, is mapped into the sequence of N_{vis} vision vectors:

$$m(V) = \{e_i\}_{i=1}^{N_{vis}}, \quad e_i \in \mathbb{R}^{d_{lm}} \tag{4.1}$$

where d_{lm} is the dimension of the LM's input embeddings. From the perspective of the LM, these vector representations $\{e_i\}_{i=1}^{N_{vis}}$ are functionally equivalent to a sequence of N_{vis} tokens embedded with the LM's text embedder. Therefore, we can say that we "allocate" N_{vis} of LM's input embeddings to the image-based vision features (Figure 2.3).

Architecture. We work with two types of mapping networks, $m(.)$:

- **MLP:** relatively simple mapping network in the form of multi-layer perceptron (MLP).
- **Q-former:** a more complex, transformer-based mapping module, proposed by BLIP2 (Li et al., 2023). The architecture of the Q-former is described in Section 2.3.

Pre-training. We argue that the pre-training of the mapping module is an important step for the appropriate alignment of the vision and language modules for the downstream tasks. Therefore, we pre-train our simple MLP mapping network on the captioning task, as described in Section 4.3. After the pre-training, the mapping network is further *fine-tuned* inside our framework for the KB-VQA task. In the case of the Q-former module, we use an

already pre-trained version from Instruct-BLIP (Dai et al., 2023), hence, we do not pre-train it ourselves. The details on the integration of Q-former as a mapping module of our system are given in Section 4.4.

4.2.4 Text-based Vision Features

For the *text-based vision* features generation we closely follow the RA-VQA framework. This allows us to rigorously study the contribution of the *image-based vision* to the framework performance.

As described in Section 3.2.1, the RA-VQA utilises a set of specialised pre-trained models to perform image-to-text transformation introduced in Section 2.2.1. As a result, the input image is described with the following textual features: image caption, detected objects and their attributes, and OCR strings.

Formally, we denote the set of utilised pre-trained models with cap, obj, and ocr, for captioning, object and attributes detection, and OCR model respectively. Textual descriptions generated for an image I are grouped to form text-based vision T:

$$T = \{cap(I),\ obj(I),\ ocr(I)\} \tag{4.2}$$

The models cap, obj, and ocr are kept *frozen*.

4.2.5 Document Retriever

For the document retriever model, we closely follow the design proposed in the RA-VQA paper (Section 3.2.2). The DPR (Karpukhin et al., 2020) based document retriever accesses the external database in order to retrieve passages relevant for the answer prediction. Utilising the Equation 3.1 the documents are matched with the question and the text-based vision features. We retrieve K documents with the highest relevance score and pass them one by one to the answer generator.

4.2.6 Answer Generator

In order to utilise the strong reasoning and generative power of modern LLMs, as introduced in Section 2.1.2, we use encoder-decoder LM as our answer generator model. The answer generator aims to combine all the information retrieved by the other models in the framework to predict an answer to the given question.

In this section, we formally define the embedding prompt used to condition the answer generator and we explain the answer generation procedure.

Embedding prompt

The answer generator is conditioned on both textual and visual modalities, which are mapped into the same embedding space of the LM. The textual inputs: question Q, text-based vision T, and document D are mapped into the text embeddings using the LM embedder e_{lm}, while the output of the vision encoder $v(I)$ is mapped with the mapping network $m(.)$ (see Figure 2.3 for the general illustration). Therefore, the set of embeddings which are used to prompt the LM, for a single document D is:

$$Z = \left\{ \underbrace{e_{lm}(Q,\, T,\, D)}_{\text{text input}},\ \underbrace{m(v(I))}_{\text{visual input}} \right\} \tag{4.3}$$

We denote Z as the *embedding prompt*.

Indifference to prompt modalities. Note that this approach to LM prompting for answer generation has inherited indifference to prompt modalities. In general, the embedding prompt used can be composed of textual input only (such as text-based vision and question), image-based vision only, or both modalities. This allows easy transition from different set-ups for training or evaluation.

Answer generation

The LM model with parameters ϕ, prompted with embedding prompt Z_k (for k-th selected document), generates the answer y_k as follows

$$\begin{aligned} y_k &= \arg\max_y p_\phi(y|\, Z_k) \\ &= \arg\max_y p_\phi(y|\, e_{lm}(Q,\, T,\, D_k),\, m(v(I))) \end{aligned} \tag{4.4}$$

The answer y_k is generated for each of K retrieved documents and the best candidate $\{\hat{y}, \hat{D}\}$ is selected by the joint probability of the document retriever (with parameters θ) and the answer generator:

$$\hat{y}, \hat{D} = \arg\max_{y, D_k} p_\phi(y|Z_k)\, p_\theta(D_k|Q,T), \tag{4.5}$$

as the final output. In this way, both the confidence of the generative LM model and of the document retriever model are taken into account.

4.2.7 Training Objective

The document retriever and answer generator model are trained with joint loss as proposed in RA-VQA. We modify RA-VQA loss (Equation 3.5) to incorporate visual image representation. Namely, the first term in Equation 3.5 now depends on both vision and text embeddings represented by embedding prompt Z:

$$\begin{aligned}\mathscr{L}_{RA-VQA-Vis} = -\sum_{(Z,S)\in\mathscr{T}}\Big(\sum_{k=1}^{K}\log p_\phi(s_k^*|Z_k) \\ + \sum_{k\in P^+(Z,S)}\log p_\theta(D_k|Q,T) - \sum_{k\in P^-(Z,S)}\log p_\theta(D_k|Q,T)\Big)\end{aligned} \tag{4.6}$$

where $\mathscr{T}$ is the whole dataset, S is the set of annotator's responses, $s_k^* \in S$ is the annotated answer that is contained in the document D_k or the most popular answer among annotators if the exact match cannot be found in the document. The P^+ and the P^- are set of *helpful* and *not helpful* documents, respectively, as defined in Section 3.2.4.

4.3 Mapping Network Pre-training

The mapping network is an important module of our system because it bridges two modalities: the vision encoder and the language model. As discussed in the Section 4.2.3, we argue that pre-training of the mapping network on vision-language task is important. With this step, we aim to obtain good initialisation for further fine-tuning inside the KB-VQA framework.

Therefore, in the case of our simple MLP mapping network, we build a pipeline for pre-training on the image-captioning task. The pipeline consists of a vision encoder, which output is processed by the mapping network, and the LM that is conditioned on the mapping network output. The vision encoder and LM are kept *frozen* while the parameters of the mapping network are trained. The system illustration is given in Figure 4.3.

Task-formulation

We frame the captioning task as the conditional generation of the targeted caption $\mathbf{y} = y_1, ..., y_L$ given the input image I. The input image is encoded with frozen vision encoder $v(.)$ and mapped with the mapping network $m(.)$ into the embeddings understood by LM. An instruction prompt P, such as "`Caption this image:`", is embedded with LM embedder e_{lm} and fed to the LM encoder. Therefore, the embedding prompt Z (Equation 4.3) for the

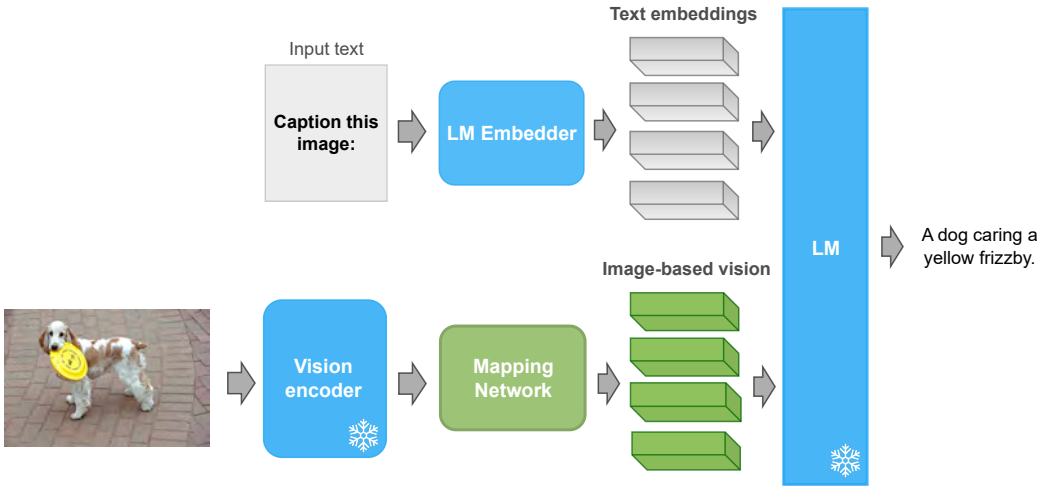


Fig. 4.3 **The MLP mapping network pre-training pipeline.** The mapping network is pre-trained on captioning task. The vision encoder and language model are frozen. The LM is provided with a task description: "Caption this image:".

captioning task is:

$$Z = \{e_{lm}(P),\ m(v(I))\} \tag{4.7}$$

Training. Considering that we keep LM frozen and that the image caption sequences are usually around one sentence long, to avoid the accumulation of error during training, we use the "teacher forcing" method (Williams and Zipser, 1989). Hence, the parameters of the mapping network are trained to maximise the likelihood:

$$\log p(\mathbf{y}|Z) = \sum_{l=1}^{L} \log p(y_l|Z, y_{<l}) \tag{4.8}$$

where the $p(y_l|Z, y_{<l})$ is the probability of the next caption token y_l given the previous caption tokens $y_{<l}$ and input embedding prompt Z. The gradients are propagated *through* the frozen LM to update the parameters of the mapping network via stochastic gradient descent.

4.4 Question Aware Visual Representations

In the previous section, we described the pre-training of the MLP mapping module. In this section, we will describe our second approach to the mapping module architecture; question-aware Q-former.

Recent work popularly chooses simple architecture for the mapping module, such as linear layer (Cho et al., 2021; Eichenberg et al., 2021; Lin et al., 2022; Tsimpoukelli et al., 2021). However, BLIP2 (Li et al., 2023) introduced the lightweight querying transformer module (Q-former) specially designed to bridge the gap between vision and language modalities. The Q-former design and pre-training technique are described in Section 2.3. Recently, InstructBLIP (Dai et al., 2023) extends on BLIP2, proposing the Q-former that can be conditioned on textual instructions, so it extracts task-relevant visual features from the frozen image encoder (Section 2.3).

4.4.1 Q-former as Mapping Module

Inspired by Q-former's performance, we integrate it into our RA-VQA-Vis framework and compare its abilities with the MLP mapping module. We fully utilise the Q-former's design by *conditioning it on the VQA question*. The question text interacts with the query embeddings through self-attention layers of the Q-former, encouraging it to extract the visual information from the frozen vision encoder that is **relevant for answering the question**. Therefore, the image-based vision features passed to the LM are no longer static per image as they were in the MLP approach; in contrast, they are specific to each image-question pair.

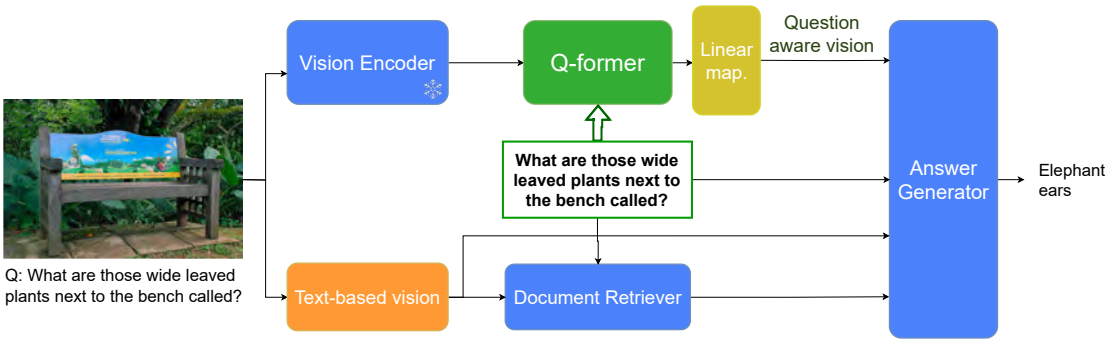


Fig. 4.4 **RA-VQA-Vis with Q-former mapping module.** The Q-former module, followed by a fully connected layer, bridges the gap between vision and language encoders. Question-aware visual representation is generated by passing the VQA question to the Q-former. This encourages the extraction of information relevant to the concrete image-question pair.

Figure 4.4 shows the RA-VQA-Vis system diagram with the mapping module $m_Q(.)$ consisting of the Q-former and a fully connected layer. In this setup, the mapping module jointly processes the vision encoder output embeddings V and the question Q to extract vision representations and map them into the LM embedding space. Concretely, $m_Q(.)$ generates

sequence of N_{vis}, d_{lm}-dimensional vision vectors as:

$$m_Q(V,Q) = \{e_i\}_{i=1}^{N_{vis}}, \quad e_i \in \mathbb{R}^{d_{lm}} \tag{4.9}$$

where d_{lm} is the dimension of the LM's input embeddings, and N_{vis} is the number of token embeddings allocated to image-based vision representations.

4.4.2 Benefits of Using the InstructBLIP Q-former

Now that we have formally defined the Q-former as a mapping module in our framework, we want to highlight the benefits of using InstructBLIP Q-former for the task of Visual Question Answering. We focus on two main advantages: potential reduction of information loss due to question-aware visual feature extraction and good weights initialisation due to extensive pre-training done by InstructBLIP authors.

Reduce of information loss

The image-based features used to represent an image can encapsulate a limited amount of information. As a result, the valuable information for answering the question may be neglected. We hope to encourage the Q-former to tailor its focus to the relevant image regions by providing it with the question. For instance, in the example from Figure 4.4, we anticipate that information propagated to the LM after the introduction of a question-aware approach will be more oriented towards the described plant, even though the image originally have details that may be considered more relevant in the general case (e.g. for image captioning task). In this way, we would effectively leverage vision encoder output embedding and hopefully enhance the image understanding segment of the KB-VQA framework.

Extensive pre-training

InstructBLIP Q-former is pre-trained on 13 different datasets covering a large number of visual-language tasks such as image captioning, visual reasoning, image classification, etc. Most importantly, it was pre-trained on the KB-VQA task. Therefore, we believe that InstructBLIP Q-former comes with good initial weights, providing a good starting point for further fine-tuning inside our system. Therefore, there is no need for us to pre-train the mapping module, which is in contrast with the MLP mapping network approach (Section 4.3).

4.5 Visual Regions of Interest

As part of our efforts to improve overall image understanding of the KB-VQA framework, so far, we have proposed the integration of an image encoder to obtain continuous image representation and the use of a question-aware mapping module to extract continuous representations relevant to the question asked. In this section, we go further and propose a question-aware procedure for the selection of relevant image *regions*.

4.5.1 Motivation

As previously discussed, the VQA question often focuses on the specific image region, which may not be the image's main focus. For example, see the image from Figure 4.5 with the question "`What candy resembles the man's tie?`". In this case, the question refers to the region covering a small ratio of the whole image. Hence the global image encoding may fail to capture enough details from it required to answer the question. To mitigate this issue, we do not only rely on global image representation, but we also form *regional* image representations. These regional features should more closely describe the relevant region, potentially including information not preserved in the global image encoding.

In the following section, we formally describe the process of including regional representations into our frameworks, and we present the algorithm for the selection of the region of interest with respect to the given question.

4.5.2 Proposed method

Object detection. For the input image I, we use off-the-shelf object detector $obj(.)$ to locate the bounding boxes $B = \{b_j\}_{j=1}^{M}$, the related object classes $C = \{c_j\}_{j=1}^{M}$, and their class confidence $T = \{t_j\}_{j=1}^{M}$ for M potential ROIs:

$$obj(I) = \{B, C, T\} \tag{4.10}$$

ROI selection algorithm. After the object detection, we select N_{ROI} regions of interest by following a procedure that prioritizes detected regions with the object class *specifically mentioned in the question*. The subsequent criteria rely on class confidence scores and region sizes. We adopt the following algorithm for ROIs selection:

1. **Occurrence in the question.** Prioritize ROIs where the object class is directly referenced in the question.

2. **Confidence treshold.** From the remaining ROIs, select the largest ones that have a confidence score surpassing threshold t.

3. **Region size.** If the number of selected ROIs is still less than required, select the rest of the objects ordered by size.

The pseudo-code of the procedure is provided in Appendix A.

After we have selected N_{ROI} regions of the image we consider relevant, the image is cropped to obtain image patches $\{r_i\}_{i=1}^{N_{ROI}}$. In the further steps, these image patches are processed in the same manner as the whole image I. Firstly, they are encoded with the vision encoder $v(.)$, and then aligned with the embeddings space of the LM using the mapping module $m(.)$. Finally, they are fed to the LM together with the textual inputs (text-based vision T, question Q, and the retrieved document D), that are previously encoded with the LM's embedder $e_{lm}(.)$. Therefore, after introduction of ROIs, the embedding prompt Z from the Equation 4.3 becomes:

$$Z = \left\{ \underbrace{e_{lm}(Q,\ T,\ D)}_{\text{text input}},\ \underbrace{m(v(I, r_1, r_2, ..., r_{N_{ROI}}))}_{\text{visual input}} \right\} \tag{4.11}$$

This embedding prompt is fed to the answer generator model to generate an answer as described in Section 4.2.6.

ROI selection example. In Figure 4.5 we show the example of 2 ROIs selected for the question "`What candy resembles the man's tie`?" Even though the provided image focus on the motorcycle, our selection algorithm chooses the image patches of the *tie* and a *man* from the background since these object classes occur in the question. We hope that providing the answer generator with the ROIs selected in this way can provide the necessary information to answer the question.

Related work. Note that REVIVE (Lin et al., 2022), and PaLM-E (Driess et al., 2023) also argue that object-centric visual representation improves image understanding and includes them in their framework. However, their region detection approaches are *question-agnostic*, unlike ours. Therefore, we hope to improve the ROIs selection algorithm by making it *question-aware* (i.e. focused on the image regions relevant for answering the VQA question), as described in this section.

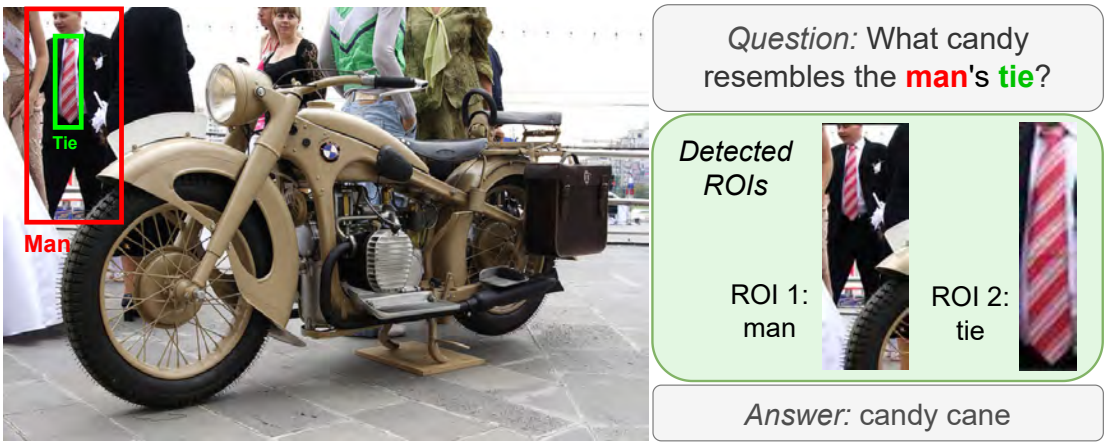


Fig. 4.5 **Question-aware detection of regions of interest.** The example of detecting 2 ROIs based on the provided question. Even though this image focus is on the motorcycle, our selection algorithm chooses the image patches of the *tie* and a *man* from the background as the ROIs for this image-question pair.

4.6 Prompt Formation

In this section, we present the template used to prompt the answer generator. Different to the RA-VQA, which use special tags to separate input features of different type (captions, attributes, documents, etc.) we use the descriptive prompt.

The prompt shown in Figure 4.6 is formed by concatenating the embeddings from the set Z (Equation 4.11) in the corresponding order. We start with the global and regional continuous image representations (`<Image>` and `<ROI>`), then we provide the task description (similar to the one used in InstructBLIP): "Use the provided image to answer the question". Furthermore, we describe the type of text-based features by "The image caption is:" and "In the image, there are the following objects", finally including the retrieved document content after "Document:".

Prompt template:

<Image> <ROI_1> <ROI_2>... Use the provided image to answer the question <Question>. The image caption is: <Caption>. In the image, there are the following objects: <Object_1>, <Attributes_1>, <Object_2>, <Attributes_2> ...
Document:<Document_content>.

Fig. 4.6 **Prompt format.** The <Image>, <ROI>, <Caption>, <Object>, <Attributes>, and <Document_content> are placeholders for the corresponding features passed to the answer generator. Each of placeholder is optional and the prompt can be modified to adjust to the absence of the specific feature.

4.7 Conclusion

In this chapter, we presented our framework for the KB-VQA task. We started by highlighting the weaknesses of the text-based vision and proposed overcoming these by introducing an image-based vision (Section 4.1).

We built our framework architecture on top of the RA-VQA baseline, extending it by incorporating the **frozen vision encoder** used for visual feature generation. Additionally, we introduce a **mapping module** to bridge the gap between the vision and language modalities.

We propose the use of two types of mapping modules: a relatively simple MLP, and a more complex, transformer-based Q-former. In Section 4.3, we detailed our approach to pre-training the MLP mapping module on a captioning task. In Section 4.4, we introduced the concept of the **question-aware Q-former**, highlighting its benefits, and describing how it will be integrated into our framework. In Section 4.5 we present the algorithm for the selection of **regional image representations** based on the question. We conclude the chapter by presenting our prompt template in Section 4.6.

In the following chapter, we give our experiment setup, specifying concrete model instances used in our work. We also describe used datasets and evaluation metrics.

Chapter 5

Experiments Setup

In the previous chapter, we provided a thorough description of our approach. Moving forward, this chapter details the experimental setup. We begin with the Section 5.1, where we describe the datasets used for the KB-VQA task and for the MLP pre-training on the captioning task. Next, although we have formally addressed our system components in the previous chapter, Section 5.2 specifies the configurations of the concrete models used for each component. In Section 5.3 we provide training parameters for our system, and with Section 5.4, we conclude the chapter by defining the VQA metric for system evaluation.

5.1 Datasets

In this section, we will describe the dataset and knowledge base we use for the KB-VQA task and the dataset used for the pre-training of the mapping network on captioning task.

5.1.1 KB-VQA Datasets

KB-VQA Dataset. We use OK-VQA dataset (Marino et al., 2019) which large proportion of questions cannot be answered only based on the image. They either require commonsense reasoning or domain-specific knowledge. The OK-VQA consists of 14,031 images and 14,055 questions split into 9,009 questions for training and 5046 questions for evaluation. We choose this dataset as it is the well-established benchmark for the KB-VQA task commonly used in related work (Section 3.1).

Knowledge Database. Following RA-VQA, we use the passage corpus collected by Luo et al. (2021) from Google Search as an outside knowledge database. This corpus originates from querying the Google Search API using questions and their corresponding

answer annotations from the OK-VQA dataset, to retrieve passages under 300 words. We use `GS-full` corpus that consists of 168,306 passages generated based on both training and test set OK-VQA questions.

5.1.2 Image Captioning Dataset

For the pre-training of the mapping network on the captioning task, we use the Conceptual Captions dataset (Sharma et al., 2018). This dataset is automatically constructed by harvesting image-caption pairs from publicly available web pages and consists of approximately 3.3M image-caption pairs

5.2 Architectural Components

In the previous chapter (Chapter 4), we described KB-VQA-Vis architecture conceptually, without specifying concrete instantiation of the components (except for the InstructBLIP Q-former). In this section, we list the specific models used in our experiments.

5.2.1 Vision Encoder

For image encoding, we use two versions of the *frozen* Vision Transformer (ViT, Dosovitskiy et al. (2020)), depending on the architecture of the mapping module.

ViT used with MLP mapping network. For our experiments with a simple MLP mapping network, we use ViT-L/14@336px version of ViT (in future denoted by ViT-L/14) pre-trained by CLIP (Radford et al., 2021) on approximately 400M image-text pairs collected from the internet (WebImageText dataset (Radford et al., 2021)) using contrastive objective (Section 2.1.1). This vision encoder has 307M parameters, and it maps input image of size 336×336 pixels to a 768-dimensional continuous image embedding. We choose this version of CLIP-ViT over the ViT-L/14@224px because it is shown that model pre-trained with higher resolution images has better performance on downstream tasks (Radford et al., 2021). The image encoder is kept *frozen*. This allows us to generate image embeddings only once, prior to the system training, which significantly reduces computational costs.

ViT used with Q-former. In the experiments with InstructBLIP Q-former we are constrained to the choice of ViT-g/14 (Zhai et al., 2022) version of ViT because this vision encoder was used during the Q-former pre-training. Any change of the vision encoder version at the time of fine-tuning can cause a drop in the performance of the Q-former. The ViT-g/14 has been pre-trained on 1B images (from ImageNet-21k corpora (Deng et al., 2009), and on

privately gathered, weakly-labelled images (Zhai et al., 2022)) for the classification task. It has an output embedding size of 1408 and 1B parameters.

A comparison of the ViT-L/14@336px and ViT-g/14 vision encoders is given in Table 5.1.

Table 5.1 **Comparison of ViT-L/14 and ViT-g/14 vision encoders.** ViT-L/14 model is used with the MLP mapping network and the ViT-g/14 is used with the InstructBLIP Q-former.

Encoder Version	Embedding dimension	Image res. (px)	# parameters (M)	Train. data (M)
ViT-L/14	768	336×336	307	400
ViT-g/14	1048	224×224	10^3	10^3

5.2.2 Mapping Module

As described in Chapter 4, we consider two different architectures of the mapping module: MLP and Q-former. Here, we describe their configurations.

MLP architecture. As described in Section 4.2.3, mapping network maps the r-dimensional image embedding into the series of N_{vis} vision tokens of dimension d_{lm} (hidden dimension of LM). For our MLP mapping network, we use a 2-layer perceptron: $\mathbb{R}^r \rightarrow \mathbb{R}^{\frac{N_{vis}d_{lm}}{2}} \rightarrow \mathbb{R}^{N_{vis}d_{lm}}$, and we reshape the output to $\mathbb{R}^{N_{vis}\times d_{lm}}$. The MLP module is pre-trained as described in Section 4.3, and further *fine-tuned* inside the RA-VQA-Vis framework.

Q-former version. We use checkpoint of InstructBLIP Q-former from HuggingFace (Wolf et al., 2020) trained as a mapping between the ViT-g/14 vision encoder and the Flan-T5-XL language model. Given that the Q-former was pre-trained on OK-VQA dataset, we *do not* fine-tune it further inside our RA-VQA-Vis framework, but we do fine-tune the linear projection layer used to align Q-former's output with the LM hidden dimension[1]. For the illustration of RA-VQA-Vis architecture with the Q-former, see Figure 4.4.

5.2.3 Answer Generator

As described in Section 4.2.6, we use LM with encoder-decoder architecture as an answer generator. This architecture choice aligns with our baseline (RA-VQA), but it excludes decoder-only models such as GPT-3 (Brown et al., 2020) and recently published LLaMA (Touvron et al., 2023) and Vicuna (Zheng et al., 2023). Following RA-VQA, we use the T5-Large (770M) variant of the T5 model family. Furthermore, we employ Flan-T5 models,

[1]The linear projection layer is initialised from the same HuggingFace checkpoint as the Q-former.

an updated version of the T5 models fine-tuned on datasets framed as instructions. Since the question-answering instruction is present in the Flan models' fine-tuning corpora, we integrated Flan-T5 models into our experiments. We use Flan-T5-Large (780M) and Flan-T5-XL (3M). Larger model variants are not tested due to computational limitations.

Training. As part of the RA-VQA-Vis framework, the answer generator model is *fine-tuned* in all of our experiments. In the case of the T5-Large, and Flan-T5-Large, we fine-tune all model parameters. For the Flan-T5-XL, we use LoRA (Low-Rank Adaptation parameter tuning, Hu et al. (2022a)) to fine-tune the model on a single GPU.

5.2.4 Document Retriever

We initialise our document retrieval model with the pre-trained version published by RA-VQA. We keep the number of retrieved documents K fixed to 5. The RA-VQA baseline showed that a further increase in the number of retrieved documents notably adds to the computational cost of the system, while it does not necessarily leads to the retrieval of additional helpful knowledge.

In most of our experiments, we keep the retriever model *frozen* since this component is not the focus of our thesis, and its fine-tuning requires extensive hyperparameter tuning (Lin and Byrne, 2022). Following the notation from RA-VQA, we use tag FrDPR (frozen document passage retrieval) to refer to the experiments in which the document retriever is frozen and tag NoDPR (no document passage retrieval) for the experiments in which document retriever is not used (i.e. the answer generator is not fed with the external knowledge).

5.2.5 Models for Image-to-Text Transformation

The central research question of this thesis is to examine whether the introduction of continuous image representations (image-based features) complements textual image descriptions. Therefore, for a fair comparison, we closely follow the RA-VQA baseline when choosing the off-the-shelf models for image-to-text transformation. The Oscar+ (Li et al., 2020c) model is used for image captioning, the VinVL (Zhang et al., 2021) model is used for detecting objects and generating their description, and the Google OCR API is used for recognition of text within the image. By selecting the same models as in RA-VQA, we can rigorously investigate the contribution of continuous image features.

5.2.6 ROIs Detection Model

As an object recognition model used for detecting potential regions of interest, we employ the VinVL (Zhang et al., 2021) because it provides us with all the necessary information: bounding boxes, object classes, and the confidence score. The threshold t used in the ROIs selection procedure is set to 0.5. We denote experiments in which we use regional image representation with an ROI tag. Hence, if not specified otherwise, the regional representation is not part of the system.

5.3 Training

Now that we listed the specific model versions used, we will give the training details for both the RA-VQA-Vis framework and for the pre-training of the MLP mapping module.

RA-VQA-Vis training. The initial learning rates are 6×10^{-5} for the answer generator, 10^{-5} for the retriever, and 3×10^{-4} for the MLP mapping module, all linearly decaying to 0 after 10 epochs. The systems with the Flan-T5-Large are trained for 8 epochs, while the Flan-T5-XL saturate after 4 epochs. The batch size was 2 with the 16 gradient accumulation steps.

MLP pre-training. The MLP module was pre-trained with a constant learning rate of 3×10^{-4}. The training is done for 15 epochs with a batch size of 64, and gradient accumulation every 2 steps.

All experiments were run on 1 Nvidia A-100 GPU with the use of Adam (Kingma and Ba, 2015) optimiser for training. The models are implemented using PyTorch and PyTorch Lightning (Falcon and The PyTorch Lightning team, 2019).

5.4 Evaluation

For the VQA evaluation metric, we use the official VQA score (Marino et al., 2019). This metric assigns the score to the generated answer based on its occurrence count in the set of human annotations S. Each OK-VQA question is annotated with 10 answers, and the VQA score is calculated as follows:

$$\text{VQA}_{score}(y) = \min\left\{\frac{\#s(y)}{3}, 1\right\}, \tag{5.1}$$

where the $s(y)$ is the count of the exact match occurrence of the generated answer y in the annotation set S. This metric ensures that the model gets partially rewarded even if it predicts the answer which is existed but is not the most popular among the human annotations.

5.5 Conclusion

In this chapter, we start with detailing the relevant datasets in Section 5.1. In Section 5.2, we provided the configurations of the concrete models for each component of RA-VQA-Vis. We then presented the training parameters in Section 5.3 and concluded with a definition of the VQA metric in Section 5.4.

After outlining the experiment setup, we provide experiment results, and discuss them in the next chapter.

Chapter 6

Results

Chapter outline. In this chapter, we present the results of our experiments. We start by positioning our top-performing model relative to the state-of-the-art solutions in Section 6.1. In Section 6.2, we provide results that test our central hypothesis: that introducing image-based features to a system that employs text-based features enhances overall image understanding.

From Sections 6.2.1 to 6.2.3, we conduct a detailed study of our KB-VQA framework. By adding each component one at a time we examine their individual and joint contributions. An ablation study of our best model is presented in Section 6.2.4. This is followed by evaluations focusing on the relevance of question-aware vision in Section 6.3, and the integration of regional image representations in Section 6.4.

Following our main experiments, Section 6.5 delves into the results of the pre-training of the MLP mapping module. We discuss the experiment regarding the number of image tokens in Section 6.5.1, and we conclude this segment by emphasizing the importance of MLP pre-training for its use in the KB-VQA framework in Section 6.5.2.

6.1 Comparison with State-of-the-Art Models

We start by comparing the performance of our framework (RA-VQA-Vis) with the top-ranked KB-VQA systems in the literature. We present our best-performing model, RA-VQA-Vis (Q-former, Flan-T5-XL) that leverages InstructBLIP's vision module and Flan-T5-XL language model to achieve **62.56**% VQA score on the OK-VQA dataset, outperforming SOTA models within its scope of parameters (< 5B), and surpassing original RA-VQA model by a large margin ($\sim$ 8% gain). The achieved VQA score positions our 4.5B parameters model at the 4$^{\text{th}}$

place of the OK-VQA dataset leaderboard[1] outperforming many systems that use very large models such as GPT-3 (175B), PaLI (17B), and Flamingo (80B) to obtain strong implicit knowledge retrieval and text generation (Table 6.1). Therefore, our results demonstrate the importance of powerful visual components and explicit knowledge retrieval that are often neglected in the literature (as discussed in Chapter 3).

Table 6.1 **RA-VQA-Vis compared to the baselines in the literature on the OK-VQA dataset.** Models are sorted by VQA score (VQA). Knowledge source abbreviations: W: Wikipedia, GS: Google Search. RA-VQA-Vis (4.5B) ranks 4th, outperforming most systems based on large models such as GPT-3 (175B). Additionally, it surpasses the original RA-VQA (underlined result) by a large margin ($\sim$ 8% gain). The RA-VQA-Vis uses the Q-former mapping module and Flan-T5-XL LM.

Rank	Model	Large Models	Knowl. Src.	# Param. < 5B	VQA
1	PaLM-E	PaLM-E (562B)	PaML-E	-	66.10
2	PaLI-X	PaLI-X (55B)	PaLI-X	-	66.10
3	PaLI	PaLI (17B)	PaLI	-	64.50
4	**RA-VQA-Vis (ours)**	**Flan-T5$_{XL}$ (4.5B)**	**GS**	✓	**62.56**
5	InstructBLIP (Vicuna-7B)	Vicuna (7B)	Vicuna	-	62.01
6	Prophet	GPT-3	GPT-3	-	61.10
7	PromptCap	GPT-3	GPT-3	-	60.40
8	BLIP-2 (Vicuna-7B)	Vicuna (7B)	Vicuna	-	59.30
9	REVIVE	GPT-3	W + GPT-3	-	58.00
10	Flamingo	Chinchilla (80B)	Chinchilla	-	57.80
11	PaLI	PaLI (15B)	PaLI	-	56.50
12	InstructBLIP (Flan-T5$_{XXL}$)	Flan-T5$_{XXL}$ (11B)	Flan-T5$_{XXL}$	-	55.50
13	BLIP-2 (Flan-T5$_{XXL}$)	Flan-T5$_{XXL}$ (11B)	Flan-T5$_{XXL}$	-	54.70
14	KAT-Ensemble	T5-large, GPT-3	W + GPT-3	-	54.41
15	RA-VQA	T5-Large	GS	✓	54.48
16	PaLI	PaLI (3B)	PaLI	✓	52.40
17	RA-VQA-FrDPR	T5-Large	GS	✓	51.52
18	TRiG-Ensamble	T5-Large	W	✓	50.50

In the following sections, we will present a systematic study of the RA-VQA-Vis framework.

[1]The leaderboard https://paperswithcode.com/sota/visual-question-answering-on-ok-vqa is last modified on May 29, 2023. We also include InstructBLIP and BLIP-2 performance reported on June 15, 2023.

6.2 Integration of Vision Encoder

We began this section by testing our main hypothesis: the introduction of image-based vision features into a framework that already utilizes text-based vision can enhance overall image understanding. We demonstrate our results for two groups of vision modules:

- ViT-L + MLP mapping network (MLP approach)
- ViT-g + InstructBLIP Q-former (Q-former approach)

Table 6.2 shows the results of the system with and without vision modules (vision encoder and mapping module). The text-based features (image caption, OCR, and detected objects with attributes) are fed to the answer generator, while the document retriever is not used (No-DPR). The first two rows in the table represent a baseline for this experiment since they do not include image-based features. We can see that both MLP and Q-former approaches bring value to the system and improve overall accuracy by *complementing text-based vision features* and therefore enhancing overall image understanding. The significant difference in the strength of both ViT-g over ViT-L as a vision encoder and Q-former over MLP as a mapping module is shown by the gap of $\sim 6\%$ in VQA score with Flan-T5$_{XL}$ model (last two rows).

Table 6.2 **Integration of vision encoder complements text-based vision and improves image understanding.** System performance with and without the use of vision modules (vision encoder and mapping module) for MLP and Q-former approaches. As a baseline, the answer generator is conditioned on text-based vision *only* (first two rows). Document retriever is not used in any of the experiments (No-DPR).

Mapping Module	Vision Encoder	LM	VQA
None	None	Flan-T5$_{Large}$	46.79
		Flan-T5$_{XL}$	50.72
MLP	ViT-L	Flan-T5$_{Large}$	49.13
		Flan-T5$_{XL}$	53.37
Q-former	ViT-g	Flan-T5$_{XL}$	**59.29**

Main insights. The integration of vision modules into the framework that originally used text-based vision only for image understanding improves system performance. The largest improvement ($\sim 9\%$) is obtained using Q-former with ViT-g.

In the following sections, we methodically examine the contribution of each system component by sequentially incorporating them into our experiments. We start with the

image-based vision-only system, followed by the addition of text-based features and the document retrieval module.

6.2.1 System with Image-Based Vision Only

In the previous section, we have shown that integration of vision modules into the framework that originally used only text-based vision improves overall image understanding. In this section, we want to test how powerful the system is with image-based vision only. The main advantage of an image-based vision-only system is that no set of specialised models for the generation of textual image descriptions is needed. This would significantly reduce system complexity and computational cost.

Set-up. The baseline set-up for this experiment consists of only an answer generator, which is conditioned on the OK-VQA question. We add the vision encoder and mapping module to condition the answer generator on the question and image-based features. The experiment results are given in Table 6.3.

We make the following conclusions from this experiment:

- By comparing VQA score of the system with and without image-based features, we conclude that **image-based vision significantly increases the VQA score**. This confirms that our vision encoder and mapping modules provide valuable information to the answer generator model.

- The ViT-g + Q-former approach achieves a notable VQA score of **58.09**%. This score is already comparable to state-of-the-art models that make use of additional components such as text-based vision, outside knowledge, and very large language models (e.g. GPT-3) (Table 6.1). The fact that we managed to achieve this performance only by using question and image-based features demonstrates the importance of having the powerful image understanding segment of the KB-VQA framework.

- The Q-former mapping module approach surpasses MLP by more than 10% points. This gap in the performance is somewhat expected given the Q-former's design (Section 2.3), and its benefits discussed in Section 4.4.2. Namely, the Q-former module is specially designed to bridge the gap between vision and language models. Additionally, the InstructBLIP version we are using is pre-trained on an extensive number of vision-language tasks (including KB-VQA), and it has the ability for question-aware vision feature extraction. However, the gap in the performance between the two approaches may also be credited to the significant difference in the vision encoder's number of

parameters (Table 5.1). Overall, the results from Table 6.3 confirm that the **ViT-g + Q-former vision system is more powerful than the ViT-L + MLP** one.

- When comparing the system's performance with different answer generator LMs, we notice that the performance of Flan models does not differ when they are prompted with the question only. However, when the image-based features are added, the gap in the performance of two LMs enlarges, showing that Flan-T5-XL has a better capacity for understanding vision prompt embeddings (rows 3-6 in Table 6.3).

Table 6.3 **Contribution of image-based features.** Image-based features are extracted with a vision encoder and a mapping module. Two configurations are considered: MLP + ViT-L, and Q-former + ViT-g. Answer generator LM is conditioned on the question only or on the question and vision-based features. Document retriever is not used (NoDPR).

LM	Question	Vision modules	VQA
T5-Large	✓	✗	24.88
	✓	ViT-L + MLP	**40.32**
Flan-T5-Large	✓	✗	27.78
	✓	ViT-L + MLP	**43.13**
Flan-T5-XL	✓	✗	27.26
	✓	ViT-L + MLP	47.02
	✓	ViT-g + Q-former	**58.09**

Main take-away. The ViT-g + Q-former vision system surpass ViT-L + MLP. It achieves **58.09**% VQA score demonstrating the importance of strong image understanding. This result is already comparable with state-of-the art models that make use of additional component such as external knowledge, text-based vision, and LLMs (e.g. GPT-3).

6.2.2 Contribution of Text-Based Vision

In the previous section, we have reported the performance of the system that uses only image-based vision and question to generate the answer. In this section, we add text-based features to the system, aiming to systematically compare the individual and joint contributions of these two types of image representations. The ablation study results are given in Table 6.4.

Insights for MLP approach. By comparing rows 2 and 3 (for Flan-T5-Large), and rows 6 and 7 (for Flan-T5-XL) we conclude that the system with only text-based features outperforms the ViT-L + MLP system with only image-based and no text-based features. The score gap of $\sim 3\%$ for both language models shows that the image understanding

Table 6.4 **Text-based features improve performance further.** Text-based features include: image caption, OCR, and detected objects with attributes. Image-based features are extracted with a vision encoder and a mapping module. **Powerful image representation surpasses textual descriptions.**

Row	LM	Vision modules	Text-based features	VQA
1	Flan-T5-Large	✗	✗	27.78
2		✗	✓	46.79
3		ViT-L + MLP	✗	43.13
4		ViT-L + MLP	✓	**49.13**
5	Flan-T5-XL	✗	✗	27.26
6		✗	✓	50.72
7		ViT-L + MLP	✗	47.02
8		ViT-L + MLP	✓	53.37
9		ViT-g + Q-former	✗	58.28
10		ViT-g + Q-former	✓	**59.29**

performance of ViT-L + MLP vision modules is not as good as the one obtained with the text-based features approach. However, when these two types of features are combined the VQA performance further increases (rows 4 and 8). Therefore, we conclude that image-based features bring *new information* to the system which makes them valuable and *compatible* with textual descriptions.

Insights for Q-former approach. In contrast to the MLP approach, the image representation obtained with ViT-g and Q-former surpasses the contribution of textual descriptions by large margin of $\sim 8\%$ (rows 6 vs. 9), achieving 58.28% VQA score. The combination of these features with textual description slightly increases the overall score (+1%). Therefore, we argue that if VLM has powerful vision components such as ViT-g + Q-former, **image-based vision is enough** for generating comprehensive image representations, and components for text-based feature generation can be omitted in favour of reduced system complexity and computational cost.

6.2.3 Contribution of Retrieved Documents

In the previous experiments, the answer generator was not fed with the retrieved documents (NoDPR). In this section, in addition to passing information extracted from an image (image-based and/or text-based features), we pass the retrieved documents to the LM, providing it with outside knowledge.

Table 6.5 shows the contribution of retrieved documents when they are combined with the vision features. The knowledge retriever model is kept frozen (FrDPR).

Table 6.5 **Integration of document retriever model improves performance further.** Image-based features are extracted with a vision encoder and mapping module. Two configurations are considered: MLP + ViT-L, and Q-former + ViT-g. Text-based features include: image caption, OCR, and detected objects with attributes. The documents are retrieved using the frozen document retriever model from the RA-VQA baseline (Fr-DPR).

LM	Vision modules	Text-based features	VQA	
			No-DPR	Fr-DPR
Flan-T5-Large	✗	✓	46.79	53.88
	ViT-L + MLP	✗	43.13	47.24
	ViT-L + MLP	✓	49.13	**55.71**
Flan-T5-XL	✗	✓	50.72	56.72
	ViT-g + Q-former	✗	58.28	60.47
	ViT-g + Q-former	✓	59.29	**62.16**

We make the following conclusions:

- From the first three rows of Table 6.5, we can see that addition of documents brings *significant improvement*. With the ViT-L + MLP vision modules, the best-performing system uses both types of image features and retrieved documents to obtain **55.71**% VQA score, which is $\sim 7\%$ points higher than the best No-DPR score.

- Some improvement with the inclusion of documents is naturally anticipated, given that OK-VQA questions are designed to require external knowledge for accurate answers. However, the remarkable improvement we achieve with the integration of the RA-VQA document retriever testifies to its strong ability to retrieve *relevant* documents from the database, as reported in Lin and Byrne (2022). This also solidifies the Google Search corpus as a valuable knowledge base for the OK-VQA dataset.

- The best-performing system (last row) uses the ViT-g vision transformer and Q-former mapping module to extract image-based vision. Combining it with textual image descriptions and retrieved documents, this system achieves VQA score of **62.16**%, outperforming the system without an integrated vision encoder by a margin of $\sim 5\%$ points (fourth row).

- We highlight that the gain in accuracy from NoDPR to FrDPR is higher for text-based features only system ($\sim +7\%$ points, Table 6.5 first row) than for image-based

features only system ($\sim +4\%$ points, Table 6.5 second row). This is expected since the document retriever selects documents based on the question, and textual image description (if provided). Therefore, the documents in the system that do not include text-based features are selected based on the question only, and therefore *less relevant* than documents selected based on both question and textual image description.

6.2.4 The Best Performing Model

In previous sections, we systematically added each system component one by one and conducted a detailed study of the individual component's contributions. We concluded that the best-performing system includes: the ViT-g vision encoder, InstructBLIP Q-former mapping module, Flan-T5-XL answer generator, RA-VQA document retriever, and set of models for text-based feature extraction (OCR, image captioning, and object detection). The system with these components we choose as our final framework configuration.

For this setup, we perform joint training of the answer generator and document retriever (as proposed in the RA-VQA baseline), denoting this model as RA-VQA-Vis. The ablation study of RA-VQA-Vis is given in Table 6.6. We conclude that each of the system's components is compatible with the others; thus it brings value and boosts the overall performance. After joint training, RA-VQA-Vis achieves VQA score of **62.56%**[2] outperforming RA-VQA baseline by large margin of $\sim 8\%$ and ranking 4^{th} on the OK-VQA dataset leader-board (Table 6.1).

Table 6.6 **Ablation study of RA-VQA-Vis components.** We report the ablation study for our best framework configuration, RA-VQA-Vis, consisting of ViT-g vision encoder, InstructBLIP Q-former mapping module, Flan-T5-XL answer generator, RA-VQA document retriever, and set of models for text-based feature extraction (OCR, image captioning, and object detection).

Model	Question	Image-based f.	Text-based f.	Documents	VQA
RA-VQA-Vis	✓	✓			58.09
	✓	✓	✓		59.29
	✓	✓	✓	✓	62.16
	✓	✓	✓	✓	**62.56**

Main takeaway. After summarising the results presented in the sections above, we can conclude that the introduction of a powerful image-based vision by employing a vision

[2]Note that due to limited resources, we did not perform extensive parameter fine-tuning for the joint training of document retriever and answer generator, as recommended in Lin and Byrne (2022). With careful selection of hyper-parameters, an increase in the overall performance is possible.

encoder and Q-former mapping module undoubtedly improves image understanding and boost the performance of the KB-VQA framework.

6.3 Contribution of Question-Aware Vision

In our results so far, we have presented our best-performing model and analyzed the impact of each system component. After comparing the Q-former approach (ViT-g + Q-former) with the MLP approach (ViT-L + MLP), we have concluded that the Q-former approach significantly outperforms MLP one. In this section, we analyse the Q-former's performance further by testing the actual contribution of its *question-aware* aspect. Our hypothesis is that by feeding the Q-former module with the question text, the visual features extracted are more relevant for answer generation.

Setup. To test this hypothesis, we compare our default system that employs the question-aware Q-former (i.e. Q-former conditioned on the question) with the system that uses *question-agnostic* Q-former (conditioned on fixed prompt "Caption this image: `<image>`"). For this experiment, we work only with image-based features (no text-based vision and documents). The comparison results are presented in Table 6.7.

We draw the following conclusions:

- System with the question-aware Q-former outperforms the question-agnostic approach by $\sim 1\%$. This demonstrates that question-aware Q-former do extract information relevant to the image-question pair, which the question-agnostic Q-former may not always select.
- Although the question-aware Q-former brings an increase in performance in comparison to the question-agnostic approach, this boost is slight ($\sim 1\%$). We suspect this is because the image understanding capabilities of ViT-g + Q-former vision modules are already very powerful and can generate comprehensive image representation for most of the image-question pairs from the OK-VQA dataset. Hence, only a small number of image-question pairs additionally benefit from the question-aware approach. For example, these could be samples for which the question is focused on the small regions in the image background, which may be seen as irrelevant before seeing the question.

Main takeaway. The Q-former conditioned on the question outperforms the question-agnostic Q-former by $\sim 1\%$. Therefore, although the Q-former's access to the question does bring valuable information for the answer generator, it is not the crucial feature of the Q-former + ViT-g vision system.

Table 6.7 **Question-aware approach slightly improves performance.** We compare two image-based features-only systems: a question-aware Q-former and a question-agnostic Q-former conditioned on the fixed prompt 'Caption this image: <image>', to assess the contribution of feeding the Q-former with the question.

LLM	Question-Aware Q-former	VQA
Flan-T5-XL	✗	57.16
	✓	58.28

6.4 Regions of Interest

In our experiments so far, we have used only global image representation as continuous vision features, in this section we include regional image representations (ROIs). We follow the procedure from Section 4.5 for selecting the relevant image regions and processing them further with a vision encoder and mapping module.

6.4.1 Different number of ROIs

We experiment with different numbers of ROIs extracted and passed to the answer generator *together* with the global image representation. For this experiment, we work only with image-based features (no text-based vision and documents). The results are presented in Table 6.8.

We draw the following conclusions:

- In the case of a ViT-g + Q-former vision system (rows 5-8), adding only 1 region of interest increases the VQA score by $\sim 1\%$. Further increase in the number of ROIs does not boost performance. We consider it reasonable that a strong vision system, such as ViT-g + Q-former, benefits the most from the addition of *only one* region of interest. Namely, the questions in the OK-VQA dataset often focus on one region of the image, and therefore, if the first ROI is chosen correctly (i.e. is the one relevant for the question), providing the model with additional ROIs does not bring any relevant information, but in contrast, it can diverge the focus, and negatively affect the system performance (rows 7 and 8). That being said, we consider the boost in performance after adding 1 ROI (row 6) as confirmation that our selection algorithm correctly chooses relevant regions.

- On the other side, the ViT-L + MLP system (rows 1-4) achieved the best performance with 4 ROIs. We suspect that the MLP system does not have the capability to extract

comprehensive information from global image representation, and hence it benefits from ROIs due to the general increase in the number of features they bring.

Table 6.8 **Different number of ROIs compared to image only method.** Image only method refers to use of only global image representation; ROIs method stands for a number of regional representations used *together* with global image representation. Document retriever, and text-based vision are not used. The answer generator is Flan-T5-XL.

Row	Vision Modules	Method	VQA
1	ViT-L + MLP	Image Only	47.02
1		2 ROIs	48.29
3		4 ROIs	**49.18**
4		6 ROIs	48.77
5	ViT-g + Q-former	Image Only	58.28
6		1 ROIs	**59.37**
7		2 ROIs	59.36
8		4 ROIs	59.02

6.4.2 Selection Methods Comparison

To further investigate whether the performance boost comes from an adequate selection procedure or only from the increase in the number of visual features, we compare our method with the "Evenly Split" method. Evenly Split refers to splitting the image to obtain the left and right half in case of 2 ROIs, or top-left, top-right, bottom-left, and bottom-right in case of 4 ROIs. The results are presented in Table 6.9. From this experiment, we conclude the following:

- **ViT-g + Q-former system** (rows 4-6). The ROI method outperforms both Images Only and Evenly Split methods. Using two evenly split image patches marginally increases the system performance ($\sim 0.3\%$), while adding two image patches selected with our procedure increases the performance by $\sim 1\%$. This result solidifies the ROI selection algorithm.

- **ViT-L + MLP system** (rows 1-3). In this case, the ROI approach very slightly outperforms the Evenly Split method ($+ \sim 0.3\%$). We can also notice that both Evenly Split and ROI methods are notably better than the Image Only approach ($\sim 2\%$), which was not the case for ViT-g + Q-former. Therefore, we conclude that less powerful vision system, such as ViT-L + MLP, benefits more from the increased number of

vision features in general than specifically from ROIs. This confirms our assumption from the previous experiment.

Table 6.9 **ROIs method compared with image only and evenly split methods.** Image only: global image representations; Evenly split and ROIs methods stand for use of global image representation *with* evenly split image patches or detected regions of interest, respectively. Document retriever, and text-based vision are not used. The answer generator is Flan-T5-XL.

Vision Modules	Method	VQA
ViT-L + MLP	Image Only	47.02
	4 Evenly Split	48.87
	4 ROIs	**49.18**
ViT-g + Q-former	Image Only	58.28
	2 Evenly Split	58.60
	2 ROIs	**59.36**

6.4.3 Contribution of text-based features

Finally, we examine the contribution of text-based features after including ROIs. We compare systems with and without text-based features, before and after the inclusion of ROIs. The results are presented in Table 6.10, and our conclusions are the following:

- In the first row, we see the results before integrating ROIs. As discussed in Section 6.2.2, in this case, adding text-based features improves the VQA score by $\sim 1\%$. However, after we enhanced the image-based features by including the regional representations (second row), the text-based features did not contribute to the system performance (the VQA score is the same with and without text-based vision). Therefore, we consider text-based vision redundant to image-based vision when ROIs are used.

  We explain this result as follows: the objects and their attributes that are part of textual image description are generated using the same object recognition model employed to select potential ROIs. Therefore, the system with both types of features contains:

  1. textual descriptions of all detected objects from the image,
  2. continuous representations of the subset of these objects that are recognised as ROIs (two in this case).

Therefore, if the selected ROI subset is sufficient to answer the question, the textual description of the remaining objects provided by the text-based vision may be unnecessary. Once again, this validates our selection procedure.

Table 6.10 **Contribution of text-based features after inclusion of ROIs.** Image-only approach refers to using only global image representation; Image + 2ROIs stands for using both global and regional image representation. Main conclusion: text-based features (OCR, image captioning, and object detection) do not contribute when ROIs are used. Document retriever is not used (NoDPR). The answer generator is Flan-T5-XL.

Vision Modules	Method	VQA	
		w/o text-based f.	w/ text-based f.
ViT-g + Q-former	Image Only	58.28	59.29
	Image + 2 ROIs	**59.36**	**59.36**

Main takeaways: Incorporating regional image representations improves image understanding. Selecting regions of interest based on the question is superior to using evenly divided image patches. For the Q-former system, adding text-based features (in the form proposed by RA-VQA) becomes unnecessary when using the ROI method.

With this experiment, we conclude the discussion on the results from our main line of work: enhancing image understanding of the KB-VQA system by incorporating continuous image representations (global and regional). In the next section, we will present the experiments relevant to the MLP mapping module pre-training.

6.5 Pre-training of Mapping Network

In this section, we turn our attention to the MLP mapping module. In all of the previous experiments, when used, the MLP mapping network was pre-trained by us. Here, we detail the results of the pre-training.

Setup. The two-layer MLP module is pre-trained on the captioning task, as described in Section 4.3. This step aims to obtain good starting initialisation of the mapping module for its further fine-tuning inside the KB-VQA framework. In order to learn the appropriate mapping between the image embedding outputs of the vision encoder and the LM embedding space, our MLP is pre-trained using the same vision encoder and LM configurations as in the KB-VQA framework. In these experiments, both the ViT-L vision encoder and LM model are frozen, while the MLP is trained from scratch.

The test set loss values of the Conceptual Captions dataset are given in Table 6.11 for three different language models. The image embeddings are mapped into a sequence of 10 tokens fed to the LM. The two Flan-T5 models are prompted with the "Caption this image: `<image>`", where `<image>` is a placeholder for image tokens obtained with the mapping network. The T5-Large LM is fed with only image tokens, without an instruction prompt, as this model is not instruction-finetuned in contrast to the Flan model family.

Table 6.11 **Pre-training of MLP mapping network on Conceptual Captions**. The image is described with 10 tokens, and the prompt: "Caption this image: `<image>`" is used for Flan-T5 models.

LLM	# Parameters (M)	Test loss
T5-Large	770	2.79
Flan-T5-Large	780	2.62
Flan-T5-XL	3000	2.42

Main insights. The loss values reported in Table 6.11 show that the LM with the largest number of parameters (Flan-T5-XL) has the best ability to understand the image tokens and generate the image captions. Additionally, the Flan-T5-Large outperforms T5-Large, even though the model size is similar. This confirms conclusions from Chung et al. (2022), which favour instruction fine-tuned versions of T5 models, Flan-T5.

6.5.1 Experiment with different numbers of image tokens

We experiment with the number of image tokens to test if we can improve image understanding by using more than 10 token embeddings for the image representation. The results for 10

and 16 image tokens are provided in Table 6.12. We observe reduce in the loss for 16 image tokens, concluding that more information on the image is successfully propagated to the LM in that case. The reduction in loss can also be due to the significant increase in the number of parameters of the mapping network ($\sim 2.5\times$) which may give the MLP more capacity to successfully bridge the vision encoder and LM.

However, we decide to keep the image representation to 10 tokens as a trade-off between the increased number of parameters and the reported loss decrease.

Table 6.12 **Increase in number of image tokens slightly improves performance.** Performance of two-layer MLP mapping network on Conceptual Captions. Prompt: "Caption this image: `<image>`".

LLM	Image tokens (n)	# MLP parameters (M)	Test loss
Flan-T5-Large	10	56.4	2.62
	16	140.5	2.57
Flan-T5-XL	10	218.6	2.42
	16	550.5	2.39

6.5.2 Pre-training of mapping network is important

Finally, we demonstrate the importance of mapping module pre-training. To do so, we train the KB-VQA system with the random initialised MLP network. We then compare it with the KB-VQA system performance reported in Section 6.2.1, which employs pre-trained MLP. The comparison is given in Table 6.13.

Table 6.13 **Pre-training of mapping network is important.** OK-VQA performance using only image-based features with and without pre-training of MLP mapping network (MN) on Conceptual Captions (Con. Cap.).

LM	Pre-trained MN	Con. Cap. Loss	VQA
Flan-T5-Large	✗	-	31.46
	✓	2.62	**43.13**
Flan-T5-XL	✗	-	39.58
	✓	2.42	**47.02**

Table 6.13 shows improvement of VQA score by $\sim +12\%$ points for Flan-T5-Large and by $\sim +7\%$ points for FLan-T5-XL based system, demonstrating the advantage of using pre-trained mapping module. This result confirms that our mapping network has successfully

learned the alignment between vision and language modalities on captioning task, which can be a valuable starting point for further fine-tuning in the KB-VQA system.

Key takeaway. The pre-training of the mapping module plays an important role in boosting the performance of the KB-VQA system.

6.6 Main Results

We conclude this chapter by highlighting the main conclusions from our experiments.

Integration of vision encoder:

- Our framework, RA-VQA-Vis, achieves **62.56**% VQA score on the OK-VQA dataset, surpassing RA-VQA baseline by large margin ($\sim 8\%$). Our 4.5B model outperforms many systems that use very large models, such as GPT-3 (175B), demonstrating the importance of powerful vision understanding for the KB-VQA task. (Section 6.1)
- The visual features obtained with ViT-g vision encoder and Q-former mapping module bring a significantly larger gain in VQA score in comparison to the text-based vision approach ($\sim 9\%$ gap). Combining these two image representations further improves system performance. (Sections 6.2.1 and 6.2.2)
- The addition of the document retriever component further increases the performance of the whole framework. (Section 6.2.3 and 6.2.4)

Question-aware Q-former:

- The Q-former conditioned on the question asked outperforms the question-agnostic Q-former by $\sim 1\%$. Therefore, although the Q-former's access to the question does bring valuable information for the answer generator, it is not the crucial feature of the Q-former + ViT-g vision system.

Regions of Interest:

- Including regional image representations increases the Q-former system performance by $\sim 1\%$. Using only one ROI shows to be sufficient in this case, validating our selection criteria. (Section 6.4.1)
- The ROI method outperforms Evenly Split and Image Only methods. (Section 6.4.2)
- Our experiments show that the use of text-based vision (as described in RA-VQA) is redundant after the inclusion of ROIs. (Section 6.4.3)

The MLP mapping module pre-training

- The pre-training of the mapping module plays an important role in the KB-VQA system. Using pre-trained MLP boosts the VQA score core by $\sim +12\%$ points for Flan-T5-Large and by $\sim +7\%$ points for FLan-T5-XL LM.

In the next chapter, we conclude the thesis.

Chapter 7

Conclusion

In this thesis, we have investigated the contribution of continuous image representation as an information source for the task of Knowledge-Based VQA. Firstly, we have proposed the integration of a vision encoder into the RA-VQA framework, the baseline system that originally relays only on a textual description for image understanding. Our experiments show that including continuous vision features result in a more comprehensive image representation, surpassing a text-based vision approach and significantly boosting system accuracy.

To further improve image understanding, we have proposed using *question-aware* Q-former as a mapping module rather than the commonly used MLP. The Q-former is conditioned on the given question and therefore aims to extract the visual features most relevant to answering the question asked. Our experiments show that the ViT-g vision encoder paired with the Q-former significantly outperforms the vision system with ViT-L vision encoder and relatively simple MLP mapping network. The notable difference in the performance can be credited to both the use of a more powerful vision encoder and the use of a more complex mapping module tailored for the VQA task.

Furthermore, we propose the use of *regional* image representation in addition to the global image features. We develop an algorithm for selecting the image regions of interest (ROIs) relevant to the asked question. Our experiments suggest that our region-based approach outperforms the whole image-based and evenly-split approaches, confirming that our selection procedure does retrieve relevant image regions.

In addition to our main line of work, we form the pipeline for pre-training the MLP mapping module on captioning task. In our experiments, we demonstrated that such a pre-training of the mapping module is an important step towards the successful alignment of vision and language modalities.

To conclude, we proposed the RA-VQA-Vis, a system built on top of RA-VQA and designed to improve its image understanding. Our best-performing model, RA-VQA-Vis

(Q-former, Flan-T5-XL), achieves **62.56**% VQA score on the OK-VQA dataset, surpassing our baseline (RA-VQA) by large margin ($\sim$ 8%). The RA-VQA-Vis (4.5B) outperforms many systems that use very large models, such as GPT-3 (175B), taking 4th place on the OK-VQA leader-board (Section 6.1), and therefore demonstrating the importance of powerful vision understanding for the KB-VQA task.

7.1 Future Work

Potential directions for future research include:

- **Continuous vision representations for external knowledge retrieval**. The RA-VQA-Vis uses only textual features and the question to retrieve relevant external knowledge. The document retrieval model can be enhanced by integrating continuous image representation in the DPR retrieval procedure.
- **Textual description of detected object complemented with continuous representations.** We currently follow the prompt template from Figure 4.6, where the ROIs are passed together with global image representations at the beginning of the prompt. Inspired by PaLM-E (Driess et al., 2023), we can modify the prompt to pass the textual descriptions of each object next to its accompanying continuous representation. In this way, the LM will be fed with structured data.
- **Systematic Error Analyses.** Currently, there is no automatic approach for error analysis that classifies OK-VQA examples based on their emphases such as image understanding, domain knowledge, or general reasoning. Developing an automated method to categorize the OK-VQA dataset examples in such categories would allow us to refine our understanding of the system's individual component performance.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. (2022). Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chen, X., Djolonga, J., Padlewski, P., Mustafa, B., Changpinyo, S., Wu, J., Ruiz, C. R., Goodman, S., Wang, X., Tay, Y., et al. (2023a). Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*.

Chen, X., Wang, X., Changpinyo, S., Piergiovanni, A., Padlewski, P., Salz, D., Goodman, S., Grycner, A., Mustafa, B., Beyer, L., Kolesnikov, A., Puigcerver, J., Ding, N., Rong, K., Akbari, H., Mishra, G., Xue, L., Thapliyal, A. V., Bradbury, J., Kuo, W., Seyedhosseini, M., Jia, C., Ayan, B. K., Ruiz, C. R., Steiner, A. P., Angelova, A., Zhai, X., Houlsby, N., and Soricut, R. (2023b). PaLI: A jointly-scaled multilingual language-image model. In *The Eleventh International Conference on Learning Representations*.

Cho, J., Lei, J., Tan, H., and Bansal, M. (2021). Unifying vision-and-language tasks via text generation. In *International Conference on Machine Learning*, pages 1931–1942. PMLR.

Chung, H. W., Hou, L., Longpre, S., Zoph, B., Tay, Y., Fedus, W., Li, E., Wang, X., Dehghani, M., Brahma, S., et al. (2022). Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Dai, W., Li, J., Li, D., Tiong, A. M. H., Zhao, J., Wang, W., Li, B., Fung, P., and Hoi, S. (2023). Instructblip: Towards general-purpose vision-language models with instruction tuning. *arXiv preprint arXiv:2305.06500*.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Driess, D., Xia, F., Sajjadi, M. S., Lynch, C., Chowdhery, A., Ichter, B., Wahid, A., Tompson, J., Vuong, Q., Yu, T., et al. (2023). Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*.

Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., et al. (2020). Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*.

Eichenberg, C., Black, S., Weinbach, S., Parcalabescu, L., and Frank, A. (2021). Magma–multimodal augmentation of generative models through adapter-based finetuning. *arXiv preprint arXiv:2112.05253*.

Falcon, W. and The PyTorch Lightning team (2019). PyTorch Lightning.

Gan, Z., Li, L., Li, C., Wang, L., Liu, Z., Gao, J., et al. (2022). Vision-language pre-training: Basics, recent advances, and future trends. *Foundations and Trends® in Computer Graphics and Vision*, 14(3–4):163–352.

Gao, F., Ping, Q., Thattai, G., Reganti, A., Wu, Y. N., and Natarajan, P. (2022a). A thousand words are worth more than a picture: Natural language-centric outside-knowledge visual question answering. *arXiv preprint arXiv:2201.05299*.

Gao, F., Ping, Q., Thattai, G., Reganti, A., Wu, Y. N., and Natarajan, P. (2022b). Transform-retrieve-generate: Natural language-centric outside-knowledge visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5067–5077.

Gao, F., Ping, Q., Thattai, G., Reganti, A., Wu, Y. N., and Natarajan, P. (2022c). Transform-retrieve-generate: Natural language-centric outside-knowledge visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5067–5077.

Gui, L., Wang, B., Huang, Q., Hauptmann, A., Bisk, Y., and Gao, J. (2021). Kat: A knowledge augmented transformer for vision-and-language. *arXiv preprint arXiv:2112.08614*.

Guo, D., Xu, C., and Tao, D. (2021). Bilinear graph networks for visual question answering. *IEEE Transactions on neural networks and learning systems*.

Hu, E. J., yelong shen, Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. (2022a). LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

Jia, C., Yang, Y., Xia, Y., Chen, Y.-T., Parekh, Z., Pham, H., Le, Q., Sung, Y.-H., Li, Z., and Duerig, T. (2021). Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*, pages 4904–4916. PMLR.

Jiang, H., Misra, I., Rohrbach, M., Learned-Miller, E., and Chen, X. (2020). In defense of grid features for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10267–10276.

Karpukhin, V., Oguz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., and Yih, W.-t. (2020). Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781. Association for Computational Linguistics.

Kingma, D. and Ba, J. (2015). Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Li, G., Duan, N., Fang, Y., Gong, M., and Jiang, D. (2020a). Unicoder-vl: A universal encoder for vision and language by cross-modal pre-training. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11336–11344.

Li, G., Wang, X., and Zhu, W. (2020b). Boosting visual question answering with context-aware knowledge aggregation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1227–1235.

Li, J., Li, D., Savarese, S., and Hoi, S. (2023). Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.

Li, L. H., Yatskar, M., Yin, D., Hsieh, C., and Chang, K. (2019). Visualbert: A simple and performant baseline for vision and language. *CoRR*, abs/1908.03557.

Li, X., Yin, X., Li, C., Zhang, P., Hu, X., Zhang, L., Wang, L., Hu, H., Dong, L., Wei, F., et al. (2020c). Oscar: Object-semantics aligned pre-training for vision-language tasks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX 16*, pages 121–137. Springer.

Li, Y., Liang, F., Zhao, L., Cui, Y., Ouyang, W., Shao, J., Yu, F., and Yan, J. (2021). Supervision exists everywhere: A data efficient contrastive language-image pre-training paradigm. *arXiv preprint arXiv:2110.05208*.

Lin, W. and Byrne, B. (2022). Retrieval augmented visual question answering with outside knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11238–11254. Association for Computational Linguistics.

Lin, Y., Xie, Y., Chen, D., Xu, Y., Zhu, C., and Yuan, L. (2022). Revive: Regional visual representation matters in knowledge-based visual question answering. *Advances in Neural Information Processing Systems*, 35:10560–10571.

Lu, J., Batra, D., Parikh, D., and Lee, S. (2019). Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. *Advances in neural information processing systems*, 32.

Luo, M., Zeng, Y., Banerjee, P., and Baral, C. (2021). Weakly-supervised visual-retriever-reader for knowledge-based question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6417–6431. Association for Computational Linguistics.

Marino, K., Chen, X., Parikh, D., Gupta, A., and Rohrbach, M. (2021). Krisp: Integrating implicit and symbolic knowledge for open-domain knowledge-based vqa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14111–14121.

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

Mokady, R., Hertz, A., and Bermano, A. H. (2021). Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734*.

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

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

Shao, Z., Yu, Z., Wang, M., and Yu, J. (2023). Prompting large language models with answer heuristics for knowledge-based visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14974–14983.

Sharma, P., Ding, N., Goodman, S., and Soricut, R. (2018). Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565.

Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., and Rohrbach, M. (2019). Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326.

Su, W., Zhu, X., Cao, Y., Li, B., Lu, L., Wei, F., and Dai, J. (2019). Vl-bert: Pre-training of generic visual-linguistic representations. *arXiv preprint arXiv:1908.08530*.

Tan, H. and Bansal, M. (2019). LXMERT: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5100–5111, Hong Kong, China. Association for Computational Linguistics.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Tsimpoukelli, M., Menick, J. L., Cabi, S., Eslami, S., Vinyals, O., and Hill, F. (2021). Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems*, 34:200–212.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., and Cao, Y. (2021). Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*.

Williams, R. J. and Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Le Scao, T., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

Wu, J., Lu, J., Sabharwal, A., and Mottaghi, R. (2022). Multi-modal answer validation for knowledge-based vqa. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 2712–2721.

Yang, Z., Gan, Z., Wang, J., Hu, X., Lu, Y., Liu, Z., and Wang, L. (2022). An empirical study of gpt-3 for few-shot knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3081–3089.

Yu, Z., Yu, J., Cui, Y., Tao, D., and Tian, Q. (2019). Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6281–6290.

Yu, Z., Yu, J., Xiang, C., Fan, J., and Tao, D. (2018). Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE transactions on neural networks and learning systems*, 29(12):5947–5959.

Zhai, X., Kolesnikov, A., Houlsby, N., and Beyer, L. (2022). Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12104–12113.

Zhang, P., Li, X., Hu, X., Yang, J., Zhang, L., Wang, L., Choi, Y., and Gao, J. (2021). Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5579–5588.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. (2023). Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Zhu, D., Chen, J., Shen, X., Li, X., and Elhoseiny, M. (2023). Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Appendix A

ROI Selection Algorithm

The pseudo-code of the ROIs selection algorithm is given in the Algorithm 1.

Algorithm 1 ROI Selection

Input: Potential ROIs, Question Q, Threshold T, Required number N_{ROI}
Output: Selected ROIs
Sort Potential ROIs by bounding box size in descending order
selected_ROIs = []
Prioritize ROIs with class in question Q
for each PotentialROI **do**
    if class(PotentialROI) is in Q **then**
        Append PotentialROI to selected_ROIs
    end if
end for
Select ROIs based on confidence if needed
for each PotentialROI not in selected_ROIs **do**
    if confidence(PotentialROI) $> T$ **then**
        Append PotentialROI to selected_ROIs
    end if
end for
Fill up to N_{ROI} if needed
while length of selected_ROIs $< N_{ROI}$ **do**
    Append unselected PotentialROI to selected_ROIs
end while
return selected_ROIs

Appendix B

Qualitative Comparison of Vision Systems

In Figure B.1 we show one OK-VQA example on which ViT-g + Q-former vision system outperforms ViT-L + MLP system. Even though the caption is not informative enough to answer the question, and retrieved documents (based on the question and text-based vision) provide misleading information, the system with the ViT-g vision encoder and Q-former mapping module predicts the correct answer.

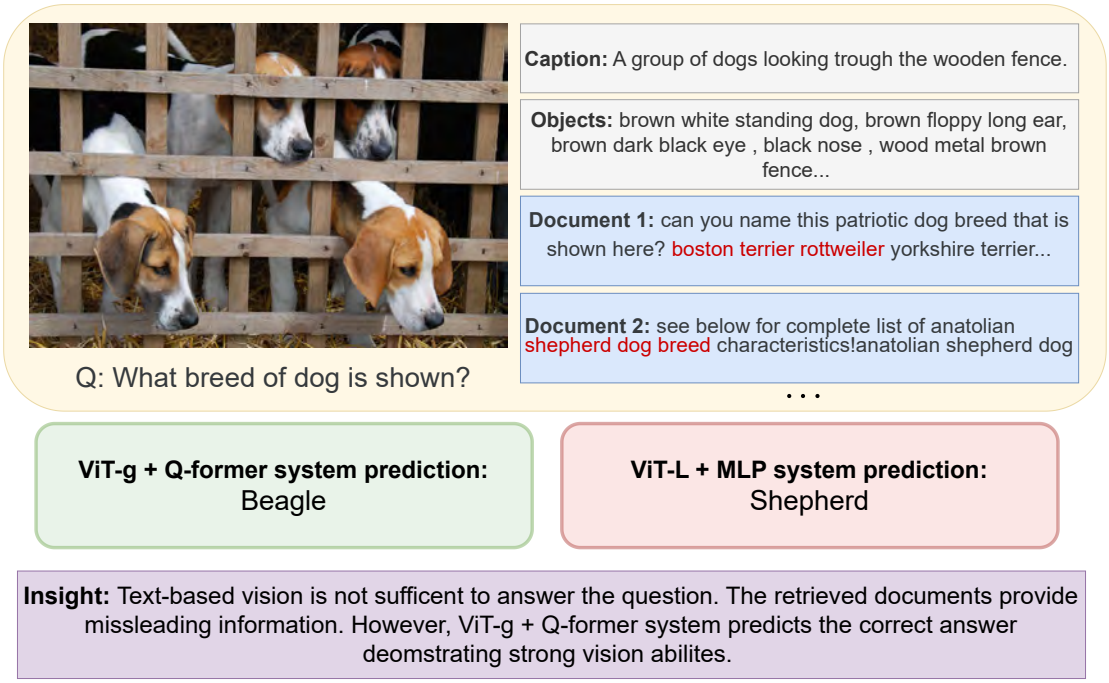


Fig. B.1 **Qualitative comparison of two proposed vision systems.** Even though the caption is not informative enough to answer the question, and retrieved documents (based on the caption) provide misleading information, the ViT-g + Q-former vision system predicts the correct answer demonstrating its superior capabilities in comparison to ViT-L + MLP system.