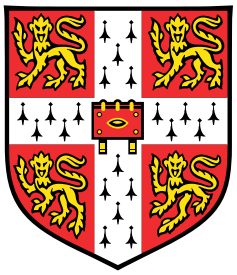
Augmenting Multi-modal Question Answering Systems with Retrieval Methods



Weizhe Lin

Department of Engineering
University of Cambridge

This thesis is submitted for the degree of
Doctor of Philosophy

Trinity College July 2024

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the preface and specified in the text. It is not substantially the same as any work that has already been submitted, or, is being concurrently submitted, for any degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee.

Weizhe Lin
July 2024

Abstract

The quest to develop artificial intelligence systems capable of handling intricate tasks has propelled the prominence of deep learning, particularly since 2016, when neural network models emerged as the mainstream approach. With applications ranging from recommender systems to speech recognition, these models have revolutionised various domains. However, challenges persist, especially in incorporating extensive domain-specific knowledge and mitigating the generation illusion inherent in large language models.

This thesis explores the integration of retrieval-augmented generation (RAG) into multi-modal question answering (QA) systems as a solution to these challenges. By leveraging external knowledge sources, RAG enhances model accuracy and access to domain-specific information. The research unfolds in the following order:

Firstly, to efficiently and effectively leverage the external knowledge for answering knowledge-intensive, visually-grounded questions, we introduce RA-VQA (Retrieval Augmented Visual Question Answering), a framework tailored for knowledge-based visual question answering (KB-VQA). We demonstrate the efficacy of joint training for retriever and generator models in maximising performance.

Secondly, FVQA (Fact-based Visual Question Answering) 2.0 introduces semi-automatically annotated adversarial samples to address data distribution imbalances and enhance system robustness, showcasing substantial improvements in handling challenging scenarios.

Thirdly, the development of FLMR (Fine-grained Late-interaction Multi-modal Retriever), a state-of-the-art multi-modal retriever, and its scaled-up version, PreFLMR (Pre-trained FLMR), underscore the significance of late-interaction models in achieving superior multi-modal retrieval performance. We show that the proposed models are capable of capturing finer-grained interactions between query and context, offering efficient and accurate retrieval across a wide range of multi-modal retrieval tasks.

Then the focus pivots to retrieval methods in TableQA, introducing ITR (Inner Table Retriever) for closed-domain scenarios and LI-RAGE (Late Interaction Retrieval Augmented Generation with Explicit Signals) for open-domain TableQA tasks. Both frameworks exhibit remarkable performance improvements over existing approaches. We show that incorporating

retrieval methods in TableQA substantially pushed the research boundary, offering state-of-the-art question answering performance.

Through meticulous experimentation and innovation, this thesis not only advances the theoretical understanding of multi-modal retrieval augmented systems but also contributes practical frameworks and datasets that address critical challenges in question answering across diverse domains. As the journey towards effective AI systems continues, these contributions serve as a solid foundation for future advancements in information retrieval and question answering in multi-modal contexts.

Acknowledgements

This Ph.D. thesis would not have been possible without the support and contributions of numerous individuals and organisations.

First and foremost, I would like to express my deepest gratitude to my supervisor, Prof. Bill Byrne. His guidance, encouragement, and relentless effort in securing the necessary funding for my research have been invaluable. Without his support, this journey would not have been possible. I am also deeply thankful to my co-supervisor, Prof. Sam Stranks. His invaluable support and the opportunity to work with the Cavendish Laboratory have been pivotal in allowing me to publish meaningful work. Their expertise and guidance have been indispensable to the completion of this thesis.

I extend my deepest gratitude to Prof. Per Ola Kristensson and Prof. Frank Keller for their invaluable contributions as examiners of this thesis. Their meticulous review and insightful suggestions have significantly enhanced the quality of this work. I am profoundly thankful for their guidance and expertise, which have not only improved this thesis but also contributed to my personal academic development.

I am profoundly grateful to my parents, whose love and unwavering support have been the foundation upon which I have built my academic career. Their sacrifices and belief in my abilities have always been my driving force.

To my girlfriend, Qiuting Wang, who has been my constant companion throughout this challenging journey, thank you for your patience, understanding, and encouragement. Your presence has been a source of strength and comfort during the toughest times.

I extend my sincere thanks to Toyota Motor Europe, the funding provider that has supported my research. Their financial assistance has been crucial in allowing me to pursue my studies and achieve my research goals.

I would also like to acknowledge the collaborative efforts of my colleagues, Kangyu Ji, Zhilin Wang, Jinghong Chen, Jingbiao Mei, Alexandru Coca, and Guangyu Yang. Working with such talented and dedicated individuals has not only been a pleasure but has also greatly enhanced the quality and impact of my projects.

I want to thank my favorite singer, Shen Zhou (Charlie). His voice has been a constant source of empowerment and motivation, cheering me up during the difficult times and inspiring me to keep pushing forward.

Lastly, I would like to express my gratitude to Margaret de Vaux, my English teacher at Trinity College. Your kindness and excellent teaching during English supervisions have significantly improved my language skills, enabling me to articulate my research more effectively.

To all those who have contributed to this journey, I am eternally grateful. This thesis is as much a testament to your support and encouragement as it is to my efforts. Thank you.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

"The Only True Wisdom is in Knowing You Know Nothing"

— Greek philosopher Socrates

1.1 Overview

Creating artificial intelligence (AI) capable of handling complex tasks has long been a goal in the field of artificial intelligence research. Since 2016, deep learning has emerged as a prominent approach in artificial intelligence, wherein neural network models are constructed and trained using data, gradually becoming the mainstream method for developing artificial intelligence [139]. These neural network models have also found widespread application in various practical scenarios in recent years, including recommender systems, conversational agents, speech recognition, sentiment analysis, and human-computer interaction. Artificial intelligence models are highly valued for their potential to tackle complex tasks such as automated analysis, logical reasoning, and content generation.

In 2021, the release of GPT-3 (Generative Pre-trained Transformer 3) [23] and a series of open-source large language models confirmed the effectiveness of expanding model parameters and training data under a data-driven approach, formally initiating a research wave focused on large models. The emergence of large language models signifies a new milestone in the field of artificial intelligence, and they have drawn attention to data-driven methods. These models, trained on massive amounts of data, demonstrate remarkable performance across various language tasks, from simple language understanding to complex text generation, exhibiting unprecedented capabilities.

Multimodality refers to the integration of heterogeneous data from diverse sources, typically encompassing language, vision, and audio information, and, in a broader sense,

extending to graph data and tabular (structured) data. Since 2021, research on multi-modal tasks has been increasingly prominent, especially as the fields of natural language processing and computer vision gradually converge. Consequently, academia and industry are investing more resources into the research of multi-modal large models. The emergence of a series of multi-modal large models led by GPT-4 [232] (such as LLaVA [198] and MiniGPT-4 [388]) signifies that large models are gradually acquiring powerful visual-language understanding and reasoning capabilities, with the potential to be applied to handle more complex multi-modal tasks. For example, strong visual-language understanding capabilities are crucial in applications like healthcare [220, 329, 371], education [18, 154], and recommender systems [203, 351], where accurate interpretation and integration of visual and textual information can significantly enhance performance and outcomes.

However, various studies [219, 44] indicate that although large models can handle various complex tasks based on their own knowledge, the knowledge they can store and proficiently apply is limited, especially with respect to complex and domain-specific knowledge. Current large models suffer from the generation hallucination [118, 196], often generating incorrect or ambiguous content when answering questions, and they struggle in scenarios requiring specialised knowledge or world knowledge. The quote from Socrates at the beginning of the chapter, "*The Only True Wisdom is in Knowing You Know Nothing*", is perfectly applicable to our expectations for good AI systems. An AI system truly possesses "True Wisdom" only when it recognises its own limitations and can leverage external resources to acquire knowledge when it lacks relevant information.

At this juncture, Retrieval-Augmented Generation (RAG), proposed during the early stage of deep learning systems, has regained attention. RAG is a pipeline that integrates information retrieval systems into the generation process of large models. It extracts the necessary knowledge from knowledge bases using retrieval systems and provides this information to the models, enabling reasoning or question answering grounded in the externally retrieved data. RAG can offer significant advantages to the generation of large models by:

- Mitigating the hallucination of large models by explicitly presenting the knowledge needed by the model, leading to more accurate answers (e.g., stock codes and product information).

- Enabling models to access a wider range of industry-specific and world knowledge. Equipping models with domain-specific knowledge bases allows them to handle specialised tasks.

- Enhancing the timeliness of models. While model training and parameter updates often occur over long periods (even months or years), information updates occur at a

much faster pace. Models extracting the latest information from continuously updated databases for question answering can significantly improve their relevance.

However, at the beginning stage of the research presented in this paper (October 2021), there were still several shortcomings in the research of multi-modal large models and multi-modal RAG systems. Multi-modal systems for knowledge-intensive tasks generally suffer from the following issues:

- Complex model structures and huge parameter sizes, yet poor performance on multi-modal tasks requiring knowledge.
- Weak performance by the employed multi-modal information retrieval systems, with low recall rates (the metric assessing the retrieval performance).
- Poor integration between the information retrieval component and the retrieval-augmented generation model, resulting in unsatisfactory answering performance even with good retrieval performance.

In the next section, we present the key research questions that our study aims to address, in order to overcome the limitations of existing multi-modal RAG systems.

1.2 Research Questions and Research Scope

Motivated by the potential advantages of RAG in enhancing AI systems, we hypothesise that large multi-modal question-answering (QA) systems can be improved through the incorporation of information retrieval methods. To investigate this hypothesis, we break down our primary objective into the following research questions:

RQ1: Can retrieval methods be utilised to enhance multi-modal systems' ability to acquire knowledge from external sources when answering domain-specific or challenging questions?

RQ2: Can retrieval methods help multi-modal systems focus more effectively on the current task by filtering out irrelevant and redundant information?

RQ3: What strategies can be employed to effectively integrate retrieval methods with multi-modal systems?

This thesis presents my research findings from 2021 to 2024 to address these research questions. To ensure both the diversity and generalisability of our proposed methodologies, this study examines two distinct applications of QA, while also accounting for the time constraints inherent in a Ph.D. program:

Visual Question Answering (VQA) systems: Focused on vision-language knowledge retrieval and retrieval-augmented generation, with a focus on solving Knowledge-based Visual Question Answering (KB-VQA) tasks (see Sec. 2.4.2 for details).

Table (structured data) Question Answering (TableQA) systems: Based on information provided in tabular data, for question answering or retrieval-augmented question answering (see Sec. 2.5 for details).

We summarise the key research findings in the final chapter of this thesis (Sec. 9.1) and discuss how they address these research questions.

1.3 System Assessment

To effectively evaluate the contributions of our proposed retrieval methods within a complete QA system, it is essential to use appropriate assessment metrics.

The most indicative measure of effectiveness is the overall QA performance, which reflects the system's ability to generate correct answers, both with and without our proposed retrieval methods. While QA performance is commonly assessed by accuracy, i.e., whether the correct answer is produced, this metric alone does not provide insight into how the retrieval methods contribute to the QA system's performance. Improvements in QA scores could be attributed to enhancements in the QA model itself, which may result from variations in training duration or random factors.

Therefore, to gain a comprehensive understanding of the integrated system comprising both retrieval and QA components, it is crucial to also evaluate the performance of the retrieval methods. Metrics such as the Recall score of the retrieval component can provide a clearer picture of its effectiveness. By analysing these metrics, we can gain insights into how well the retrieval methods contribute to the QA system's overall performance. This dual evaluation of both QA and retrieval performance enables a more nuanced understanding of how the retrieval methods enhance or impact the QA process, ultimately leading to more accurate assessments of their contributions to the system. In this thesis, we will employ various approaches to assess the retrieval performance both qualitatively and quantitatively to ensure that our proposed approaches are truly effective.

1.4 Contributions and Outline

The thesis is split into four main parts: First, a thorough introduction of relevant research background is presented to define the challenges being investigated (Chapter 2). Second, Chapter 3-6 present the research work on developing systems for KB-VQA. Next, Chapter

7-8 focus on developing closed-domain and open-domain TableQA systems augmented by retrieval models. Finally, Chapter 9 summarises the findings and contributions of this dissertation, and sheds light on potential future research directions.

We introduce the research questions in Chapter 1. Subsequently, each of the chapters from 3 to 8 presents evidence and solutions for two or three of these questions. Finally, we summarise the key findings and comprehensively address these questions in Chapter 9.

Below we summarise the contributions of each chapter:

Chapter 2: Background

This chapter systematically reviews the existing approaches in literature related to the research presented in this thesis. It starts with reviewing Large Language Models (LLMs) and Large Multi-modal Models (LMMs), information retrieval, and RAG systems in a wide perspective, and then moves on to introduce methods dedicated to multi-modal RAG for VQA and TableQA.

Chapter 3: Retrieval Augmented Visual Question Answering Framework

This chapter introduces RA-VQA (Retrieval Augmented Visual Question Answering), a framework for performing RAG on multi-modal knowledge-intensive tasks that requires access to outside knowledge. In the framework, the retriever and the answer generator are jointly optimised during training to maximise the performance of both. This research work was presented in the publication:

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* (EMNLP 2022).

Chapter 4: Introducing Adversarial Samples into Fact-based Visual Question Answering

This chapter investigates the creation of a new dataset, Fact-based Visual Question Answering (FVQA) 2.0, that adds adversarial samples to the original FVQA dataset [311]. This dataset mitigates the data distribution imbalance of FVQA with a semi-automated annotation scheme. We show that systems trained with FVQA 2.0 can be more robust to adversarial samples. This research work was presented in the publication:

Lin, W., Wang, Z., and Byrne, B. (2023). FVQA 2.0: Introducing adversarial samples into fact-based visual question answering. In *Findings of the Association for Computational Linguistics* (EACL 2023).

Chapter 5: Fine-grained Late-interaction Multi-modal Retrievers

In this chapter, we proposed a novel multi-modal retriever, FLMR (Fine-grained Late-interaction Multi-modal Retriever), based on late interaction. It achieved state-of-the-art

retrieval performance on prominent KB-VQA tasks. This research work was presented in the publication:

Lin, W., Chen, J., Mei, J., Coca, A., and Byrne, B. (2023). Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering. In *Thirty-seventh Conference on Neural Information Processing Systems* (NeurIPS 2023).

Chapter 6: Scaling Up Fine-grained Late-interaction Multi-modal Retrievers

This chapter introduces a powerful general-purpose foundation retriever model, PreFLMR (Pre-trained FLMR), which was trained on millions of multi-modal retrieval data points and achieves strong performance across 9 knowledge-intensive tasks. This research work was presented in the publication:

Lin, W., Mei, J., Chen, J., and Byrne, B. (2024). PreFLMR: Scaling up fine-grained late-interaction multi-modal retrievers. In *Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics* (ACL 2024).

Chapter 7: An Inner Table Retriever for Robust Table Question Answering

This chapter shifts the topic to retrieval methods applied to TableQA. We propose Inner Table Retriever (ITR), a general-purpose retrieval method for handling long tables in TableQA by extracting sub-tables to preserve the most relevant information for a question. We show that ITR can be easily integrated into existing systems to improve their accuracy. This research work was presented in the publication:

Lin, W., Blloshmi, R., Byrne, B., de Gispert, A., and Iglesias, G. (2023). An inner table retriever for robust table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (ACL 2023).

Chapter 8: Late-Interaction Retrieval for Table Question Answering

This chapter introduces LI-RAGE (Late Interaction Retrieval Augmented Generation with Explicit Signals), which applies late interaction retrieval models and a joint training scheme of the retriever and reader to Open-domain TableQA tasks. The combined strategies set a new state-to-the-art performance on two public open-domain TableQA datasets. This research work was presented in the publication:

Lin, W., Blloshmi, R., Byrne, B., de Gispert, A., and Iglesias, G. (2023). LI-RAGE: Late interaction retrieval augmented generation with explicit signals for open-domain table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (ACL 2023).

Chapter 9: Conclusions

This final chapter concludes the contributions of this thesis, answers the research questions, and proposes promising future research directions based on my observations and insights.

1.4.1 Publications during the Time of Study

Below is a compilation of papers that I have authored or co-authored throughout my Ph.D. studies. This list includes works initiated prior to the commencement of the Ph.D. programme and completed during the course.

- [183] **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* (EMNLP 2022).
- [185] **Lin, W.**, Shou, L., Gong, M., Pei, J., Wang, Z., Byrne, B., and Jiang, D. (2022). Transformer-empowered content-aware collaborative filtering. In *KaRS@ RecSys*, pages 53–64 (KaRS 2022).
- [184] **Lin, W.**, Shou, L., Gong, M., Pei, J., Wang, Z., Byrne, B., and Jiang, D. (2022b). Combining unstructured content and knowledge graphs into recommendation datasets. In *KaRS@ RecSys*, pages 45–52 (KaRS 2022).
- [33] Chen, J., **Lin, W.**, and Byrne, B. (2023). Schema-guided semantic accuracy: Faithfulness in task-oriented dialogue response generation. *arXiv preprint arXiv:2301.12568*.
- [128] Ji, K., **Lin, W.**, Sun, Y., Cui, L.-S., Shamsi, J., Chiang, Y.-H., Chen, J., Tennyson, E. M., Dai, L., Li, Q., Frohna, K., Anaya, M., Greenham, N. C., and Stranks, S. D. (2023). Self-supervised deep learning for tracking degradation of perovskite light-emitting diodes with multispectral imaging. Nature Machine Intelligence.
- [58] Coca, A., Tseng, B.-H., **Lin, W.**, and Byrne, B. (2023). More robust schema-guided dialogue state tracking via tree-based paraphrase ranking. In *Findings of the Association for Computational Linguistics* (EACL 2023).
- [189] **Lin, W.**, Wang, Z., and Byrne, B. (2023). FVQA 2.0: Introducing adversarial samples into fact-based visual question answering. In *Findings of the Association for Computational Linguistics* (EACL 2023).
- [186] **Lin, W.**, Blloshmi, R., Byrne, B., de Gispert, A., and Iglesias, G. (2023). An inner table retriever for robust table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (ACL 2023).
- [187] **Lin, W.**, Blloshmi, R., Byrne, B., de Gispert, A., and Iglesias, G. (2023). LI-RAGE: Late interaction retrieval augmented generation with explicit signals for open-domain table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (ACL 2023).

- [57] Coca, A., Tseng, B.-H., Chen, J., **Lin, W.**, Zhang, W., Anders, T., and Byrne, B. (2023). Grounding description-driven dialogue state trackers with knowledge-seeking turns. In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue* (SIGDIAL 2023).

- [188] **Lin, W.**, Chen, J., Mei, J., Coca, A., and Byrne, B. (2023). Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering. In *Thirty-seventh Conference on Neural Information Processing Systems* (NeurIPS 2023).

- [338] Yang, G., Chen, J., **Lin, W.**, and Byrne, B. (2024). Direct preference optimization for neural machine translation with minimum Bayes risk decoding. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (NAACL 2024).

- [34] Chen, J., **Lin, W.**, Mei, J., and Byrne, B. (2024b). Control-DAG: Constrained decoding for non-autoregressive directed acyclic t5 using weighted finite state automata. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (NAACL 2024).

- [190] **Lin, W.**, Mei, J., Chen, J., and Byrne, B. (2024). PreFLMR: Scaling up fine-grained late-interaction multi-modal retrievers. In *Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics* (ACL 2024).

- [217] Mei, J., Chen, J., **Lin, W.**, Byrne, B., and Tomalin, M. (2024). Improving hateful memes detection via learning hatefulness-aware embedding space through retrieval-guided contrastive learning. In *Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics* (ACL 2024).

Chapter 2

Background

This chapter introduces the background of several topics that are related to the research work in this thesis. This chapter is organised as follows:

Sec. 2.1 offers insights into the evolving landscape of Large Language Models (LLMs) and Large Multi-modal Models (LMMs).

Sec. 2.2 explores recent advancements in Information Retrieval.

Sec. 2.3 delves into Retrieval Augmented Generation techniques.

Sec. 2.4 introduces Visual Question Answering (VQA). Recent systems and models are introduced and discussed.

Sec. 2.5 examines Table Question Answering (TableQA) and recent model developments for both closed-domain and open-domain scenarios.

2.1 Large Language Models and Large Multi-modal Models

In this section, we start with an introduction to the Transformer architecture that has been extensively used in recent deep-learning models (Sec. 2.1.1). We will then briefly introduce the development of Large Language Models (LLMs) (Sec. 2.1.3) and Large Multi-modal Models (LMMs) (Sec. 2.1.4).

2.1.1 Prerequisite: Transformer Models

The Transformer architecture [307] represents a pivotal advancement in natural language processing (NLP), profoundly impacting diverse tasks such as machine translation, text generation, and language understanding. Pre-trained language models founded on the

Transformer paradigm, exemplified by BERT (Bidirectional Encoder Representations from Transformers) [66] and GPT-2 (Generative Pre-trained Transformer 2) [245], have significantly expanded the frontiers of research concerning language models. In recent years, Transformer architectures have showcased their potential by successfully tackling tasks extending beyond language processing. Notably, Vision in Transformers (ViT) [72] has demonstrated comparable, if not superior, performance compared to traditional convolutional neural networks (CNNs) [158, 144] in vision-centric tasks.

The Transformer model introduces an innovative architecture aimed at replacing the traditional recurrent or convolutional layers, relying solely on self-attention mechanisms. The Transformer architecture comprises two primary components: the encoder and the decoder, each comprising a stack of identical layers. These layers are interconnected through multi-head self-attention mechanisms and position-wise fully connected feed-forward networks, facilitating parallelisation and efficient training, as depicted in Fig. 2.1.

Attention Mechanism

The self-attention mechanism, which lies at the heart of the Transformer model, replaces the sequential nature of Recurrent Neural Networks (RNNs) [282] with a fully parallelisable structure. This allows the model to compute attention over all positions in a sequence simultaneously, greatly improving computational efficiency and reducing training times, especially when handling large datasets.

At its core, the attention mechanism is designed to dynamically weigh the importance of different tokens in a sequence relative to one another. Specifically, the input to each attention layer consists of three vectors: Queries (Q), Keys (K), and Values (V). Each of these vectors is derived from the input embeddings through learned linear projections. The output of the attention mechanism is computed as a weighted sum of the value vectors (V), with weights determined by the compatibility of the corresponding query (Q) and key (K) vectors. Mathematically, this is expressed as:

$$Attention(Q,K,V) = \textbf{softmax}(\frac{QK^T}{\sqrt{d_k}})V,$$

where d_k represents the dimensionality of the keys and queries. The use of the scaling factor $\frac{1}{\sqrt{d_k}}$ helps mitigate the risk of having overly large dot product values, which can lead to small gradients when applying the softmax function. The self-attention mechanism allows each token in the input sequence to attend to every other token, capturing both local and global dependencies within the data.

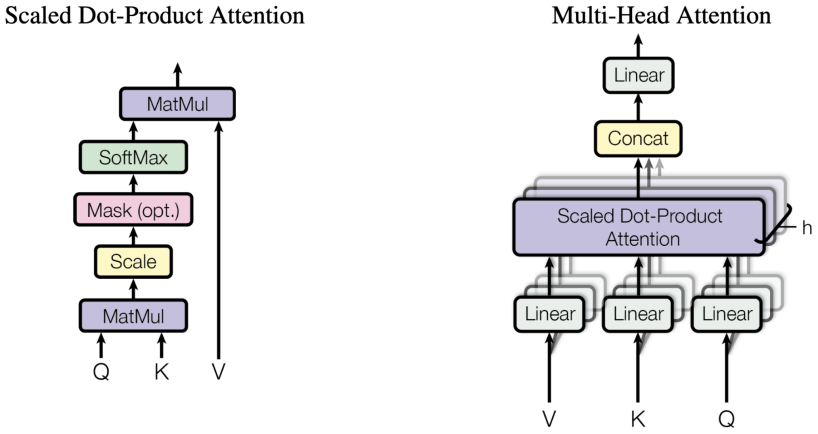


Fig. 2.1 (left) Scaled Dot-Product Attention. (right) Multi-Head Attention consists of several attention layers running in parallel. The figure is from Vaswani et al. [307].

Multi-head attention extends this idea further by running multiple attention mechanisms (or "heads") in parallel, with each head operating on a different learned projection of Q, K, and V. The outputs from all attention heads are then concatenated and linearly transformed to produce the final output for the layer. This allows the model to jointly attend to information from different representation subspaces, enhancing its ability to capture complex dependencies in the data.

Positional Encoding

One of the key innovations of the Transformer model is the incorporation of positional encodings to compensate for the lack of inherent sequential information in the self-attention mechanism. Since the Transformer does not rely on recurrence or convolution, it requires a way to inject information about the relative positions of tokens in a sequence. This is achieved through positional encodings, which are added to the input embeddings.

The positional encodings are constructed using sine and cosine functions of different frequencies. For each position pos in the sequence and each dimension i of the embedding, the positional encoding is defined as:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

Here, d_{model} is the hidden dimension of the model. This formulation allows the model to learn relative positional relationships between tokens, enabling it to generalise to sequences of varying lengths. Moreover, the periodic nature of sine and cosine ensures that the positional

encodings are continuous and smoothly vary across positions, aiding the model in learning sequential patterns.

Encoder and Decoder

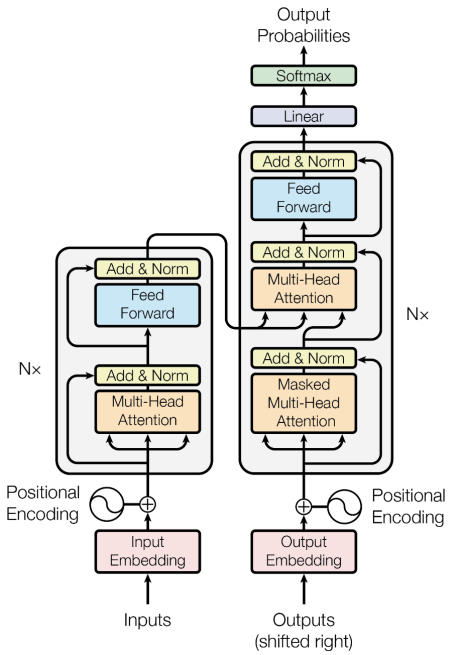


Fig. 2.2 Basic transformer architecture encoder (left) and decoder (right). Both the encoder and the decoder are composed of N blocks. Each block in the encoder includes a Multi-Head Attention module and a Feed Forward module, detailed further in the main content. The decoder's blocks are similarly structured but feature an additional Multi-Head Attention module that integrates output features from the encoder. Positional Encoding is utilised to capture the positional information of the input tokens, as elaborated in the main content. The figure is from Vaswani et al. [307].

There are two types of basic transformer blocks: **encoder** and **decoder**. The encoder is composed of N identical layers. Each layer consists of two sub-modules with residual connections between input and output: (1) a Multi-Head Attention (as described before) and; (2) a simple Feed Forward network. Encoder blocks conduct self-attention to extract features from inputs.

A decoder generates the output token based on the already generated sequence while attending to the input sequence with attention mechanisms. Therefore, the output sequence is shifted right by 1 and is then presented to the decoder. To ensure that the predictions for position i can depend only on the known outputs at positions less than i, the embeddings of the output sequence are partly masked to prevent subsequent positions (later than i in

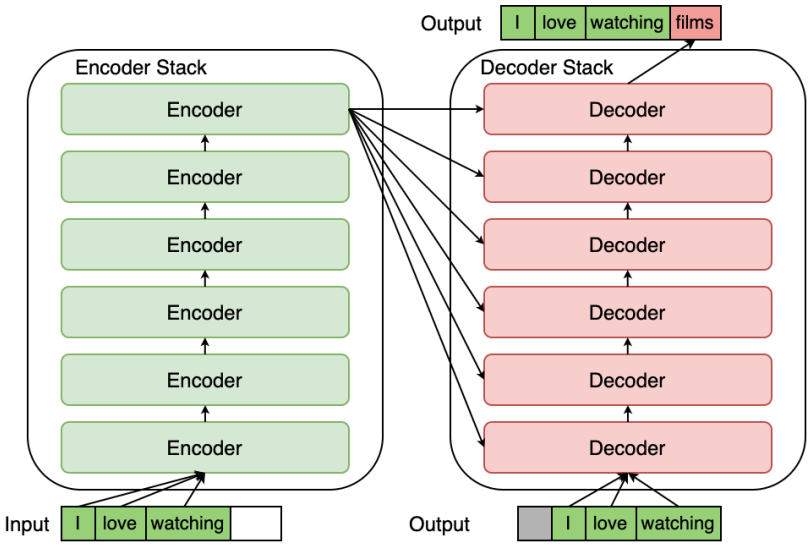


Fig. 2.3 An abstract demonstration of the encoder-decoder transformer described in Vaswani et al. [307]. This figure demonstrates how the next token "films" is predicted using the encoder-decoder transformer. The figure is from Lin [182].

output) from being used. The decoder is further incorporated with a third sub-module, as shown in the right half of Fig. 2.2. This new sub-module scores the features from the already generated output sequence with the input sequence, which exploits the information of the input sequence. The N encoders and N decoders of the original transformer are stacked as in Fig. 2.3.

Moreover, the Transformer integrates positional encodings to convey the sequential arrangement of words within the input sequence, compensating for the absence of inherent sequence order within the self-attention mechanism. Typically represented as sine and cosine functions of varying frequencies, these positional encodings augment the input embeddings, facilitating the model's ability to discern the positional context of words in the sequence.

The encoders and decoders may be utilised jointly or separately, giving rise to three distinct variants of Transformer models that are extensively investigated within contemporary research discourse:

Encoder-only Transformer: In the encoder-only Transformer, the architecture is solely composed of encoder layers. This variant is commonly employed in tasks where only the input sequence needs to be processed without generating an output sequence directly. Encoder-only Transformers are widely used in tasks such as text classification, where the model only needs to encode the input text into a fixed-size representation for classification. BERT is a prominent example of an encoder-only Transformer. BERT was pre-trained on large text corpora using unsupervised learning tasks, such as masked language modeling

and next sentence prediction, and then fine-tuned for various downstream tasks like text classification and named entity recognition (NER).

Beyond language tasks, encoder-only Transformers are widely used as a feature extractor. For example, they are used to extract features from images for image classification; they are employed to obtain dense representations (dense vectors) for users and items to be recommended in recommender systems.

Decoder-only Transformer: Conversely, the decoder-only Transformer consists solely of decoder layers. This variant is employed in tasks where the model generates an output sequence based on some given context or conditioning information. They are commonly used in autoregressive tasks, such as language modeling and text generation, where the model predicts the next token in a sequence based on the preceding tokens. Recent pre-trained LLMs like GPT (Generative Pre-trained Transformer) [244] and its successors (GPT-2, GPT-3 [23], and the current state-of-the-art GPT-4 [232]), LLaMA [303] (LLaMA2 [304]), and Mistral [132], are all examples of decoder-only Transformers. These models are trained using autoregressive objectives to generate coherent and contextually relevant text, making it a perfect match to chat and interact with users as an AI assistant, which promises wide applications across the industry.

Encoder-Decoder Transformer: The encoder-decoder Transformer combines both encoder and decoder layers into a single architecture. This variant is employed in tasks involving sequence-to-sequence transformations, where an input sequence is mapped to an output sequence. They are widely used in machine translation, text summarisation, and conversational modeling, where the model needs to understand the input sequence and generate a corresponding output sequence. The original Transformer model, introduced by Vaswani et al. [307], utilises the encoder-decoder architecture for translation tasks. This architecture is also used in some recent pre-trained models, such as T5 (Transfer Text-to-Text Transformer) [248] and BART [166].

However, decoder-only models (such as LLaMA [303] and GPT-4 [232]) are more appreciated by the community recently. One of the underlying reasons is that decoder-only models already demonstrated superior performance, and they can be highly scalable and parallelisable, as each token in the output sequence can be generated independently given the preceding tokens. This enables efficient training and inference on parallel computing architectures, leading to faster model development and deployment.

In summary, these variants showcase the versatility of the Transformer architecture, allowing it to adapt to different tasks by modifying the configuration of encoder and decoder components. Each variant comes with its unique strengths and applications, contributing to the widespread adoption and success of Transformer-based models in various domains.

2.1.2 Training Transformers

In this section, we provide a brief overview of the optimizers, schedulers, and relevant background information used for training Transformers.

Optimizers

We use Adam [149] and AdamW [204] optimizers in this study. Since the technical details of these algorithms are beyond the scope of this thesis, readers are encouraged to consult the original papers for comprehensive formulas and in-depth explanations.

The Adam optimizer, short for Adaptive Moment Estimation, is widely used for training neural networks due to its efficiency and effectiveness. It computes adaptive learning rates for each parameter based on first and second moment estimates of the gradients. Adam calculates moving averages of the gradient (first moment) and the squared gradient (second moment), correcting them to account for their bias towards zero, particularly in early training stages. This results in more reliable updates and faster convergence.

In contrast, AdamW modifies Adam by decoupling weight decay regularisation from the gradient update process. Traditional Adam applies weight decay as a penalty directly to the loss function, which can lead to suboptimal updates. AdamW, however, applies weight decay directly to the weights, independently of gradient-based updates. This approach preserves the benefits of regularisation while ensuring efficient and stable optimisation, improving generalisation in deep learning models.

Random Seeds

In deep learning model training, a random seed is an initial value used to initialise the random number generator. This generator influences various stochastic processes within the training pipeline. By setting a random seed, these random processes can be reproduced, ensuring that the training process is deterministic and can be replicated. This is crucial for debugging, comparing different models, and validating results, as it allows researchers and developers to achieve the same results consistently across different runs.

Also, setting different random seeds can be used to verify the robustness of model training. By training the model multiple times with different random seeds, we can observe how sensitive the model's performance is to the initial conditions and random processes.

Epochs

An epoch is a complete pass through the entire training dataset. Training a neural network typically involves multiple epochs, allowing the model to learn and refine its weights over time. Each epoch includes multiple iterations, each processing a 'batch' of data.

Learning Rate

The learning rate is a critical hyperparameter that determines the step size the optimizer takes while minimising the loss function. A higher learning rate can accelerate training but may overshoot the optimal solution, resulting in suboptimal outcomes. Conversely, a lower learning rate offers more precise updates but increases training time. Selecting the appropriate learning rate involves experimentation, influenced by the complexity of the model and the characteristics of the data.

Batch Size

Batch size is the number of training samples used in one forward and backward pass. Smaller batch sizes provide more updates per epoch and can lead to faster convergence due to frequent gradient updates. However, they may produce noisier gradient estimates, affecting training stability. Larger batch sizes offer more stable gradient estimates but require more computational resources and can slow convergence due to fewer updates per epoch. Choosing the right batch size balances computational efficiency and training stability.

Gradient Accumulation

Gradient accumulation addresses hardware limitations that restrict batch size by effectively increasing it. Instead of updating the model's parameters after each batch, gradients are accumulated over several batches before a single update is performed. This simulates the effect of a larger batch size, enhancing stability and performance without needing additional memory.

Schedulers

Schedulers adjust the learning rate during training. Two common schedulers are:

Constant Scheduler: Keeps the learning rate fixed throughout the training process. This simple approach can be effective for smaller datasets or models where a single learning rate suffices for the entire training duration. This strategy is always a reasonable choice, especially when the total number of required training steps is uncertain.

Linear Scheduler: Gradually decreases the learning rate from an initial value to a lower value over the training period. This helps achieve finer convergence by reducing the learning rate in a controlled manner, allowing for smaller and more precise adjustments in the later training stages.

2.1.3 Large Language Models

In this sub-section, we briefly discuss the recent advances of Large Language Models.

Early Pre-trained Large Language Models

"Large Language Models", in today's context, typically refers to language models with billion-scale parameters. In 2018 and 2019, BERT [66] and GPT-2 [245] opened the door to large pre-trained language models. BERT is among the most widely used encoder-only language models. It significantly pushed the boundaries of state-of-the-art on a wide range of language tasks, inspiring the development of many subsequent models built upon it. For example, RoBERTa [391], ALBERT [157], DeBERTa [105], XLM [59], XLNet [342], UNILM [70], and other encoder-only models followed this path in the following years. GPT-2, one of the most widely used decoder-only language models, was pre-trained on a large collection of webpages. It demonstrated that, with pre-training, language models can achieve good zero-shot performance on various language tasks. BERT and GPT-2 together laid the foundation of LLMs.

In 2019, the introduction of T5 [248] demonstrated the promising future of a unified framework that casts all NLP tasks as a text-to-text generation task, powered by large-scale pre-training. Its successors have also showcased strong capabilities across numerous tasks, such as mT5 (multi-lingual T5) [336], T0 [262], and FLAN-T5 [55].

Instruction Tuning

Soon after the release of T5, training language models with instruction tuning became a popular trend in this field. The training objectives of LLMs (aiming to minimise next-word prediction errors) typically do not align with the ultimate goal of interacting with users to "follow their instructions helpfully and safely." Consequently, instruction tuning garnered significant attention to align the model output with users' task instructions; the model was trained on (instruction, desired output) pairs to enable the model to respond to various tasks instructed by users. Over the following years, numerous instruction-tuning datasets were released. A series of pre-trained models were instruction-finetuned to achieve

notable performance in multitasking and instruction-following. Some notable models include InstructGPT (instruction-finetuned from GPT-3)[234], BLOOMZ[224] (derived from BLOOM [266]), FLAN-T5 [55] (based on T5 [248]), Alpaca [298], and Vicuna [51] (derived from LLaMA [303]), along with ChatGLM2 [75] (derived from GLM [75]).

Prompt Engineering

A prompt is the initial input or query given to an LLM to generate a response. It sets the context for the model's output, guiding it to produce relevant and coherent text based on the provided information. The prompt can include instructions, questions, or specific data, influencing the content and direction of the generated text. Thanks to the enhanced instruction-following capability of LLMs, prompt engineering became feasible and emerged as a new popular technique. Unlike previous paradigms where models undergo retraining or fine-tuning for specific downstream tasks/domains, prompt engineering [199, 302] provides a mechanism to 'fine-tune' model outputs through carefully crafted instructions [260, 29], thereby activating the intrinsic intelligence of pre-trained LLMs with prompt designs. This efficient adaptation empowers LLMs to excel across diverse tasks and domains without necessitating retraining and fine-tuning [245, 260].

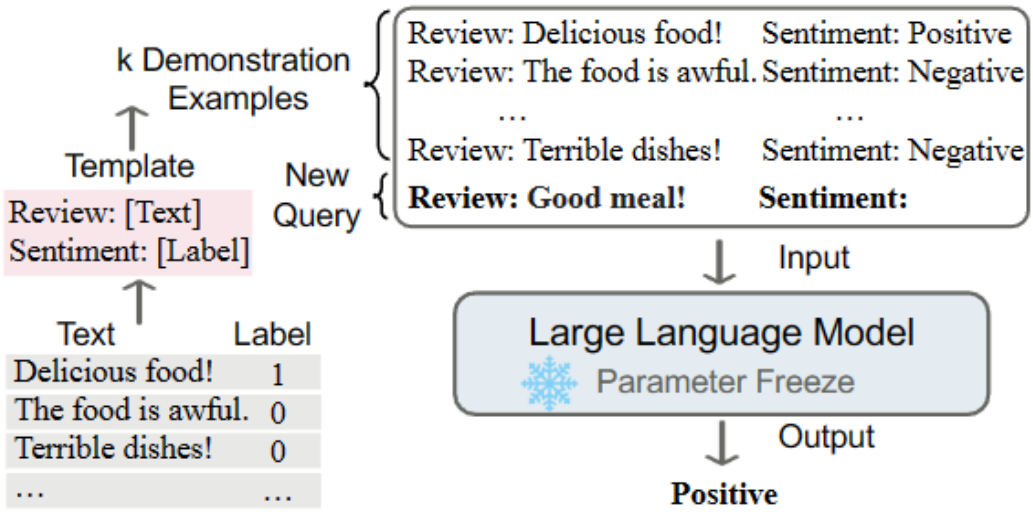


Fig. 2.4 Illustration of In-Context Learning. Providing demonstration examples similar to the query will substantially improve LLM's ability to generate correct responses. The example in the figure shows how in-context examples can be used in sentiment analysis. The figure is from Dong et al. [71].

There are various methods for prompting LLMs. Few-shot prompting [23] involves presenting examples of similar tasks prior to posing the question. This approach has evolved

into in-context learning, extensively used to train the model to stick to provided examples [71, 334]. For example, in Fig. 2.4, similar examples with ground-truth labels are provided prior to the query. The LMM is more capable of generating correct responses by referring to the provided *in-context* examples.

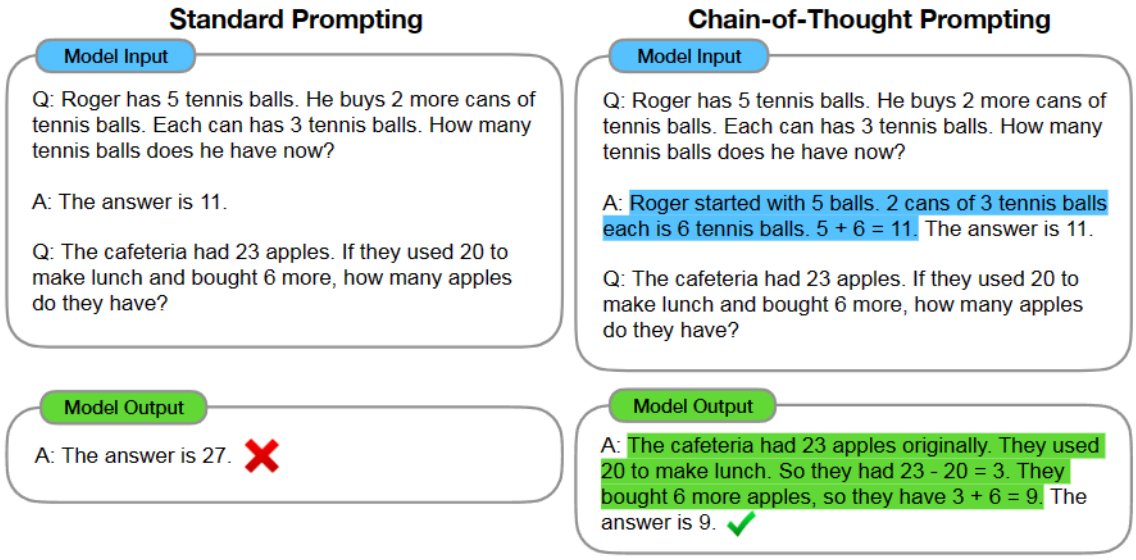


Fig. 2.5 Illustration of Chain-of-Thought (CoT) prompting. CoT enables LLMs to tackle problems through a series of intermediate steps prior to providing a final answer. This approach significantly enhances the problem-solving capabilities of LLMs by emulating the human reasoning process. The figure is from Wei et al. [320].

Chain of Thoughts (CoT) [320, 372, 313, 377] stands out as one of the most prominent prompting schemes. CoT approaches prompt LLMs to think and reason step by step, significantly enhancing their reasoning abilities. As demonstrated by Fig. 2.5, breaking down a task into multiple logical intermediate steps elicits more structured and thoughtful responses from LLMs compared to traditional prompts. Subsequent research [111, 347, 348, 387, 317] further extends this concept by exploring different reasoning structures and planning schemes, such as Tree of Thought [347], Graph of Thought [348], and Thread of Thought [387].

Alignment

Alignment is the process of steering AI systems towards human goals, preferences, and principles [223]. Though instruction tuning is effective in aligning model output with user instructions, LLMs can still exhibit unintended behaviours.

To improve the alignment of the model and avoid unintended behaviors, RLHF (Reinforcement Learning from Human Feedback) [54, 234] was used to create ChatGPT and

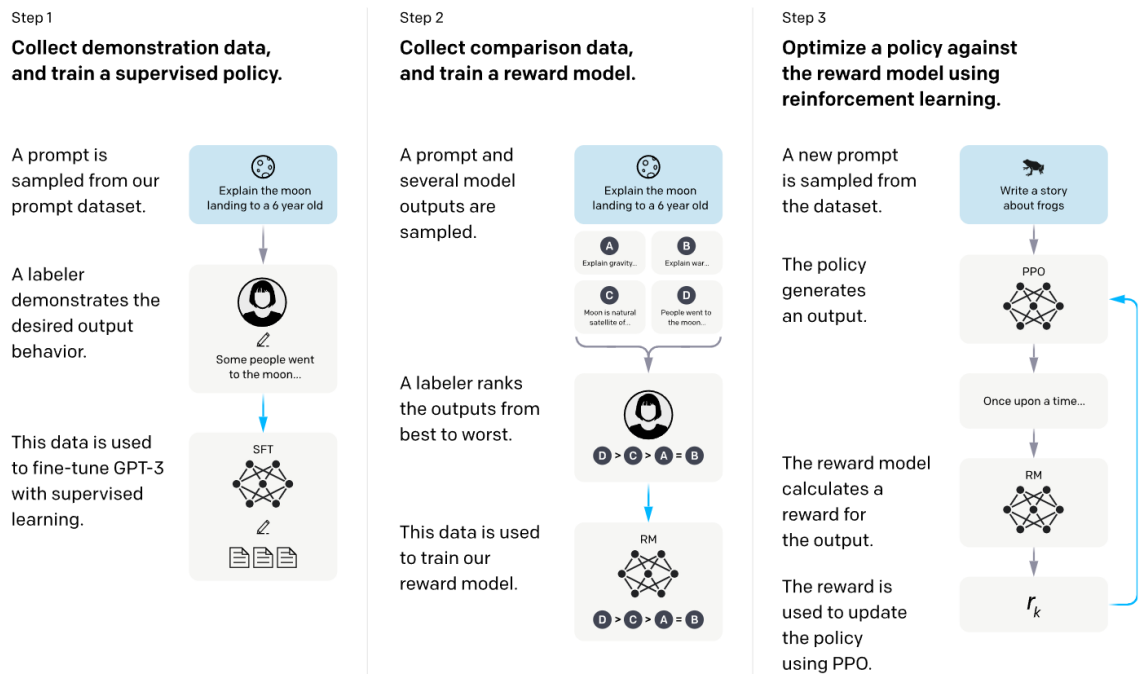


Fig. 2.6 The process of RLHF (Reinforcement Learning from Human Feedback) used in training GPT models [234]. Initially, an LLM trained via supervised learning generates several different responses. Human annotators subsequently re-order these responses. This reordered data is utilised to train a reward model, which in turn is employed to align the model with human preferences through reinforcement learning. The figure is from Ouyang et al. [234].

GPT-4, which are currently the state-of-the-art LLMs. As shown in Fig. 2.6, RLHF first trains a reward model on alignment preferences annotated by humans, and the reward model learns to rate and score the different outputs from a pre-trained LLM. This feedback signal is leveraged to fine-tune the LLM with the help of Proximal Policy Optimization (PPO) [270], an optimisation algorithm in reinforcement learning. As a result, the LLMs are more capable of generating contents that align with human preferences.

More recent investigations endeavour to achieve alignment without resorting to the reinforcement learning paradigm, aiming to enhance the computational efficiency of alignment training. Traditional reinforcement learning-based approaches often rely on complex reward models and extensive computational resources, making them less practical for scalable alignment solutions. Recently, Direct Preference Optimisation (DPO) [247] eliminates the necessity for a reward model while still achieving notable alignment performance. As shown in Fig. 2.7, DPO directly optimises the model's parameters based on preference data, bypassing the need for a reward model. This method not only simplifies the alignment process but also reduces the computational overhead associated with training and deploying

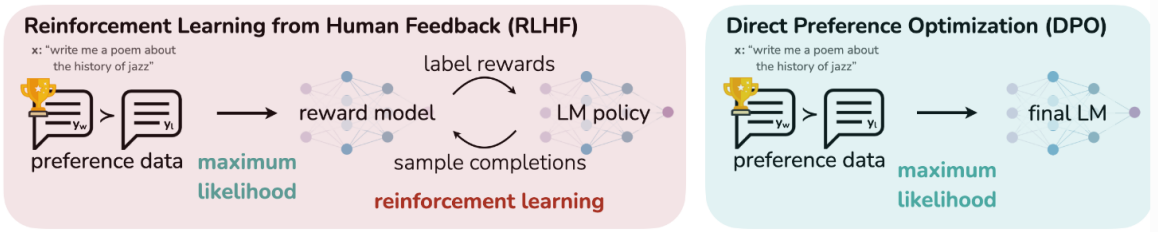


Fig. 2.7 Direct Preference Optimization (DPO): A novel approach that directly optimises model parameters based on preference data, eliminating the need for a reward model and enhancing computational efficiency in alignment training. The figure is from Rafailov et al. [247].

a separate reward model in reinforcement learning. By leveraging direct feedback, DPO can efficiently adjust model behaviours in line with human preferences, offering a streamlined and effective approach to alignment. On the other hand, Kahneman-Tversky Optimisation (KTO) [79] demonstrates that alignment can be attained even in the absence of paired data containing human preference annotations. This approach leverages binary feedback data (e.g., upvote/downvote) to infer human-like preferences, reducing the dependency on large datasets of explicit paired human feedback.

These advancements signify a paradigm shift in alignment strategies, transitioning from traditional reinforcement learning towards more direct and resource-efficient methodologies. These approaches represent promising directions in the pursuit of scalable and effective AI alignment, particularly valued by industrial companies, thereby paving the way for future research and development in this critical domain.

Recent Large Language Models

More recently, while the general training scheme remains unchanged, typically following the sequence of pre-training, instruction fine-tuning, and alignment, LLMs have entered a phase of vigorous development. With GPT-4 setting the benchmark, a succession of open-source and closed-source LLMs have emerged, closely trailing the performance of GPT-4. The number of model parameters has escalated from billion-scale to trillions. Notable examples of LLMs include Claude 3 [3], Vicuna, Mistral, LLaMA 1/2, Gemini [300], Grok [327], and PaLM 1/2 [53, 10].

Parameter-efficient Fine-tuning

Another crucial technique for utilising LLMs is parameter-efficient fine-tuning. Low-Rank Adaption (LoRA) [110], along with its successors (such as QLoRA [65] and DoRA [201]),

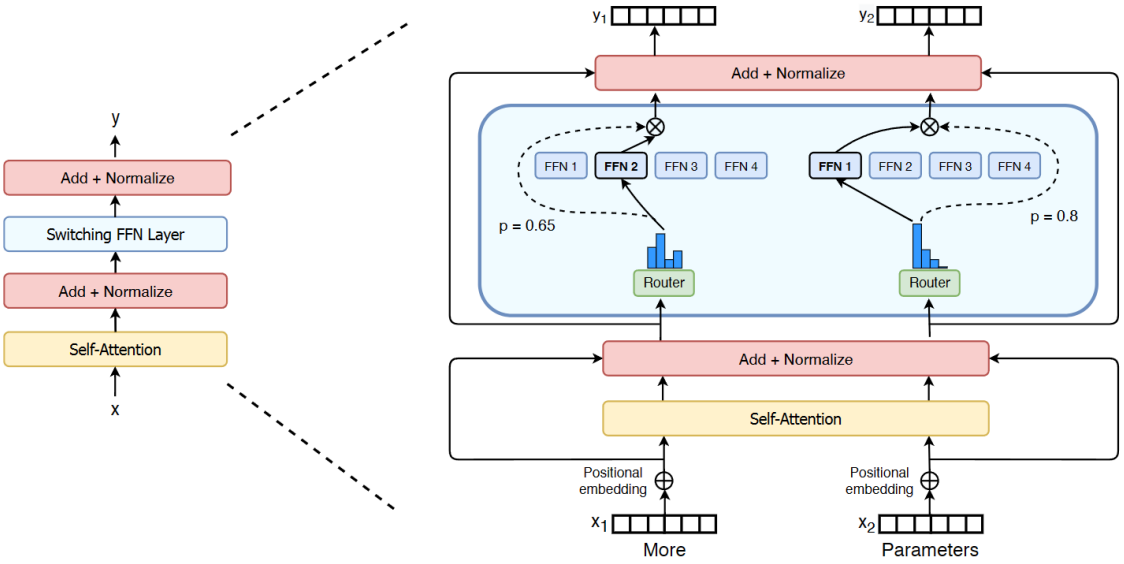


Fig. 2.8 Switch Transformer [84], a Mixture-of-Expert (MoE) model. It utilises an MoE approach to achieve efficient and scalable model performance. By dynamically selecting a subset of experts for each input, the Switch Transformer significantly reduces computational costs while maintaining or even improving accuracy compared to traditional transformer models. This architecture enables training of much larger models without proportional increases in computational resources. The figure is from Fedus et al. [84].

is now widely employed in fine-tuning LLMs. These methods introduce additional weights and freeze the original parameters of LLMs to rapidly adapt a model to fine-tuning data without the need for extensive computational resources, while still achieving comparable performance to full-parameter fine-tuning.

For example, as depicted in Fig. 2.9, LoRA optimises the adaptation process by introducing low-rank decomposition to the weight updates. LoRA decomposes the weight updates into low-rank matrices, specifically representing the update as $W + \Delta W$, where $\Delta W = A \times B$ and A and B are much smaller matrices. This approach significantly reduces the number of trainable parameters from $d \times k$ to $r \times (d + k)$, where d is the input feature dimension, k is the output feature dimension, and r is the LoRA attention dimension (the "rank"), typically much smaller than d and k. LoRA can be integrated into existing transformer-based models by inserting the low-rank update matrices A and B into specific layers, such as attention layers or feed-forward networks, without modifying the model's original architecture. During fine-tuning, only the low-rank matrices A and B are updated while the original weights remain frozen, enabling efficient adaptation to new tasks with minimal computational overhead. Despite the reduction in the number of trainable parameters, LoRA maintains or even improves the performance of the fine-tuned model by capturing task-specific information

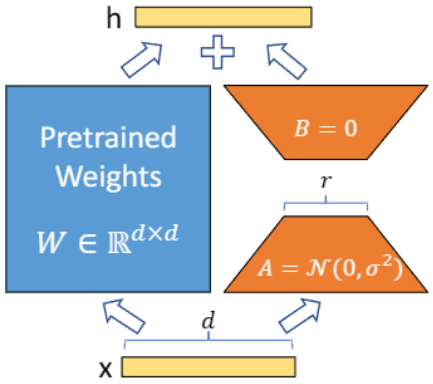


Fig. 2.9 Schematic of LoRA (Low-Rank Adaptation) [110] applied to an LLM, illustrating the integration of low-rank update matrices A and B into specific layers, reducing the number of trainable parameters while maintaining model performance. The pre-trained weights can also be $d \times k$ if the input and output feature dimensions need to be different. The figure is from Hu et al. [110].

through the low-rank updates without overfitting. This makes LoRA particularly beneficial for adapting LLMs to specific domains or tasks, offering a scalable and resource-efficient solution for fine-tuning. Given that training LLMs typically demands thousands of GPUs (Graphics Processing Units), LoRA methods are preferred by researchers and developers operating under constrained compute budgets.

2.1.4 Large Multi-modal Models

This section will briefly introduce the recent development of Large Multi-modal Models. Since this thesis primarily addresses vision-language tasks, this section will concentrate on vision-language LMMs.

Model Architecture

Constructing models capable of jointly understanding multi-modal signals and performing tasks in real-world scenarios is a long-standing research challenge. Though LLMs exhibited emergent abilities in instruction following, In-Context Learning, and Chain-of-Thoughts as described in the previous section, they remain unable to process other modalities such as vision and speech. Conversely, pre-trained Large Vision Models (LVMs)[150, 104, 281, 369, 233], including ResNet[104], excel in understanding visual inputs and performing vision tasks, such as image segmentation and object detection. However, these models were not designed to perform language tasks such as reasoning, and are unable to handle complex

vision and language tasks [354]. Therefore, a proper design is necessary to enable models to process and comprehend multi-modal inputs effectively.

Modern Large Multi-modal Models (LMMs) are predominantly built upon LLM backbones, integrating pre-trained multi-modal models (such as LVMs) with LLMs to facilitate joint understanding of multi-modal inputs. These models are also referred to as "Multi-modal Large Language Models (MLLM)" [354, 368]. Specially, in this thesis, the term LMM generally refers to MLLM, as MLLM represents the most prevalent form of advanced LMM.

A typical LMM configuration comprises three components: a pre-trained LLM, a pre-trained modality encoder, and a connector/mapping network that links them. The pre-trained LLM is responsible for understanding and reasoning, while the modality encoder processes multi-modal features, such as visual features. The mapping network aligns these components, enabling the LLM to jointly interpret different modalities.

Commonly-used Modality Encoder

Recent LMMs frequently employ pre-trained visual encoders to derive features from images. The most commonly utilised visual encoders include Vision Transformers (ViT) and its variants, such as ViT [72], CLIP-ViT [246], EVA-CLIP ViT [292], and OpenCLIP ViT [50]. Convolution-based visual encoders are also viable, such as OpenCLIP-ConvNext-L [49] as utilised in Osprey [364].

These visual encoders were pre-trained on vision-centric datasets, including LAION-2B [269], WIT [289], and COYO-700M [27]. For instance, the widely adopted CLIP-ViT [246] is a component of the CLIP model, which incorporates a ViT [72] visual encoder and a text encoder, trained to align the representations of images with their corresponding texts. After pre-training, the visual encoder in CLIP effectively extracts semantic features from images and can be decoupled to serve as the visual encoder in LMMs.

Many popular visual encoders [50, 246] were pre-trained on images with relatively low resolutions (e.g., 224×224 or 336×336 in pixels), limiting their ability to capture fine-grained details in images. Recent studies [15, 197, 176, 216] have demonstrated that using higher resolution images significantly enhances performance. Some works [15, 197] have replaced low-resolution visual encoders with those capable of processing high-resolution images. Another approach [176, 191] involves partitioning high-resolution images into smaller patches to enable detailed visual analysis.

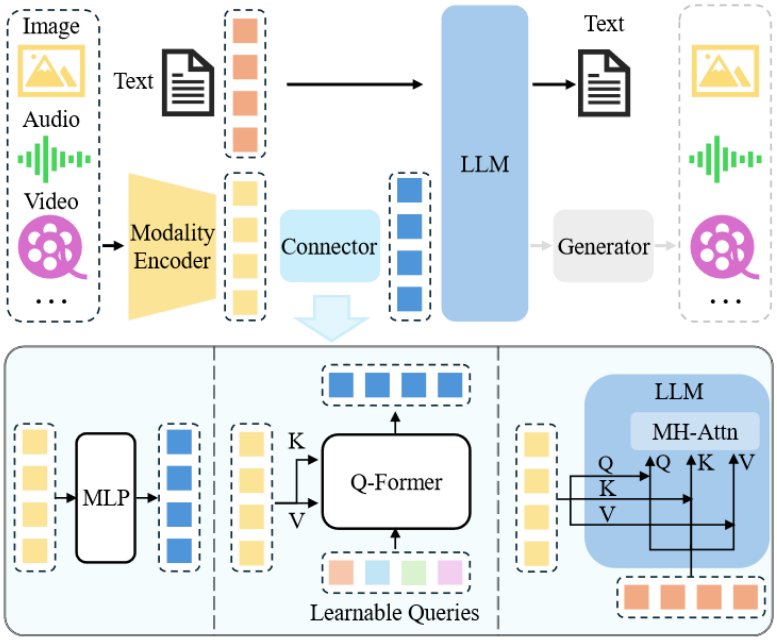


Fig. 2.10 A figure for illustration is borrowed from Yin et al. [354] to demonstrate a typical MLLM architecture. It normally comprises an encoder, a connector/mapping network, and an LLM. Optionally, a generator may be integrated with the LLM to produce additional modalities beyond text. The encoder ingests images, audio, or videos and outputs features, which are subsequently processed by the connector to enhance the LLM's comprehension. There are generally three categories of mapping networks: projection-based (e.g. MLP), query-based (e.g. Q-Former), and fusion-based (e.g. Cross-attention). The first two types employ token-level fusion, converting features into tokens that are transmitted alongside text tokens, whereas the last type facilitates feature-level fusion within the LLM.

Modality Fusion

The mapping network (or connector, projector) is trained to align the features of other modalities with the latent feature space of the text model (LLM). As shown in Fig. 2.10, the mapping network can be a simple, straightforward linear layer (or MLP, Multi-layer Perception), which has been proved to be highly effective for building many recent advanced LMMs. Some LMMs leverage more complicated design, such as Q-Former [171] (used in BLIP2 [171], InstructBLIP [62], and MiniGPT-4 [388]), P-Former [131] (used in DLP [131]), MQ-Former [205] (used in Lyrics [205]), and Cross-attention (used in IntrenVL [46], QWen-VL [15], and Flamingo [7]). Q-Former and Cross-attention are extensively used in recent LMMs. Below we elaborate further on these two methodologies.

Q-Former, first introduced in BLIP2, trains a few learnable query tokens as query features to extract image features with Transformer blocks. The connector (Q-Former) is trained

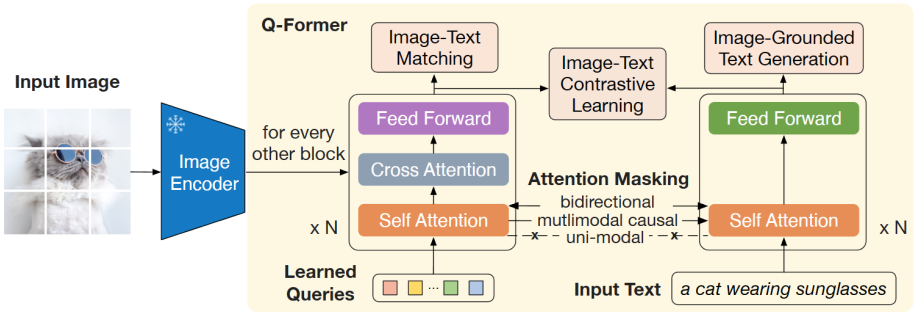


Fig. 2.11 The architecture of Q-Former [171] and the first stage pre-training. The model is tasked with three objectives: Image-Text Matching, Image-Grounded Text Generation, and Image-Text Contrastive Learning. Post training, the Q-Former is capable of understanding image content. The figure is from Li et al. [171].

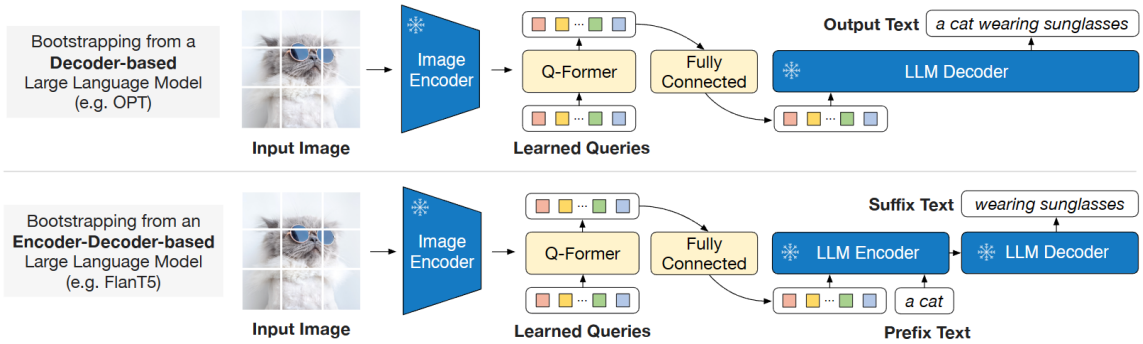


Fig. 2.12 The second stage training of BLIP2 with Q-Former [171]. The Q-Former is connected to the language model and is optimised to handle multi-modal tasks. The embeddings of query tokens are fed to LLMs to generate responses relevant to the input image and instruction. The figure is from Li et al. [171].

with two stages. In the first stage, as shown in Fig. 2.11, the connector is tasked with three objectives: (1) Image-Text Contrastive Learning encourages the model to correctly pair images with their corresponding text descriptions, helping align the visual and textual representations; (2) Image-Text Matching involves a binary classification task where the model predicts whether a given image and text pair match, improving the association between visual features and textual information; (3) Image-Grounded Text Generation requires the model to predict the masked words based on the visual context. These tasks equip the Q-Former with abilities to understand image and produce visual embeddings that can be processed by the language model. In the second stage, as demonstrated in Fig. 2.12, the Q-Former is integrated with either OPT or FLAN-T5 and subsequently fine-tuned to handle multi-modal tasks. The Q-Former receives both the image and the text instruction as input. It processes the visual features derived from the image encoder in conjunction with the text

instruction. It utilises the trained query vectors to extract visual features from the image encoder that are relevant to the given text instruction. The output produced by the Q-Former consists of visual features that have already incorporated the information from the text instruction. These instruction-aware visual features are then fed to the LLM, which generates a response or output based on these features and the provided instruction.

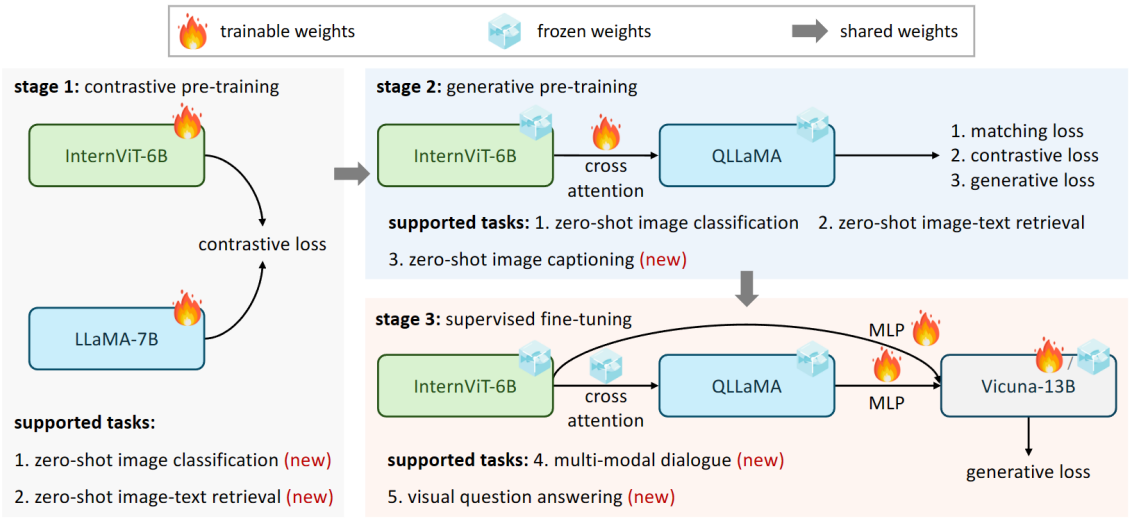


Fig. 2.13 InternVL [46] uses cross-attention to integrate features from the image encoder with the Language Middleware (QLLaMA). The model undergoes three distinct training stages, each with specific objectives, to cultivate a robust vision-language understanding capability. QLLaMA, an 8 billion parameter model, serves as a bridge facilitating communication between the scaled-up vision transformer (InternViT-6B) and the language model, enhancing overall performance on vision-language tasks. The figure is from Chen et al. [46].

The architecture of Cross-attention is less complicated relative to Q-Former. Recent LMMs use cross-attention to integrate features of different modalities. For example, InternVL (Fig. 2.13) trains Cross-attention to align the visual encoder with a Language Middleware, which are then combined with Vicuna-13B to generate responses. QWen-VL (Fig. 2.14) trains a connector that takes in learnable query token embeddings and attends to image features with cross-attention.

Mainstream LMMs

Mainstream LMMs are often built upon mainstream LLMs, leveraging the advancements and foundational capabilities of the latter to enhance their multi-modal functionalities. This integration allows LMMs to efficiently process and generate outputs based on diverse inputs such as text, images, and other data types. Prominent examples of such integrations include Flamingo [7], BLIP2 [171], LLaVA [198], MiniGPT-4 [388], InstructBLIP [62], PALI-X [39],

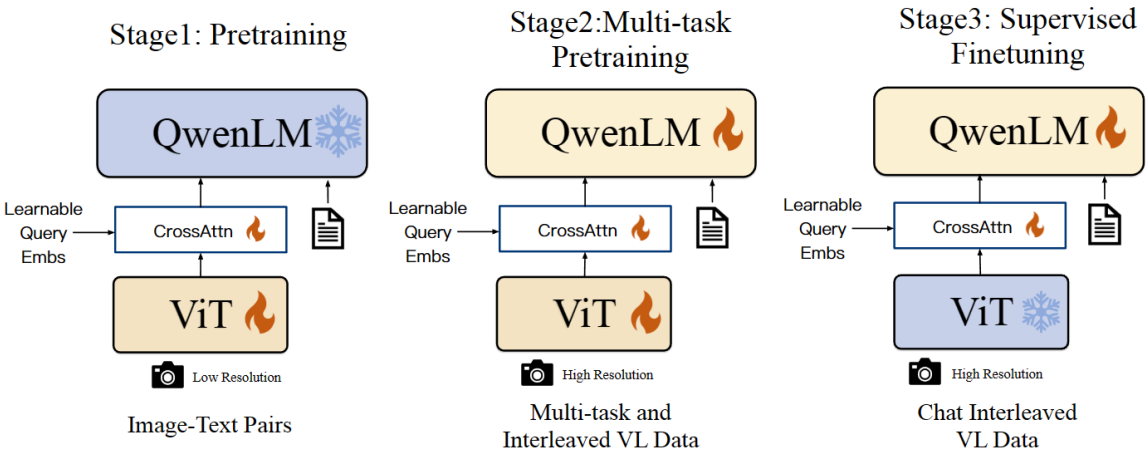


Fig. 2.14 QWen-VL [15] employs cross-attention mechanisms to synthesise features from the image encoder. The model undergoes three distinct training stages, each with specific objectives aimed at aligning the visual input with the language model. In both the second and final stages, the model is trained using interleaved VL data. This approach equips the model with the ability to manage multi-image inputs, conduct multi-round dialogues, engage in multilingual conversations, and perform fine-grained visual recognition. The figure is from Bai et al. [15].

QWen-VL [15], MiniGPT-5 [380], LLaVA 1.5 [197], MiniGPT-v2 [35], and InternVL [46], each showcasing unique features and strengths.

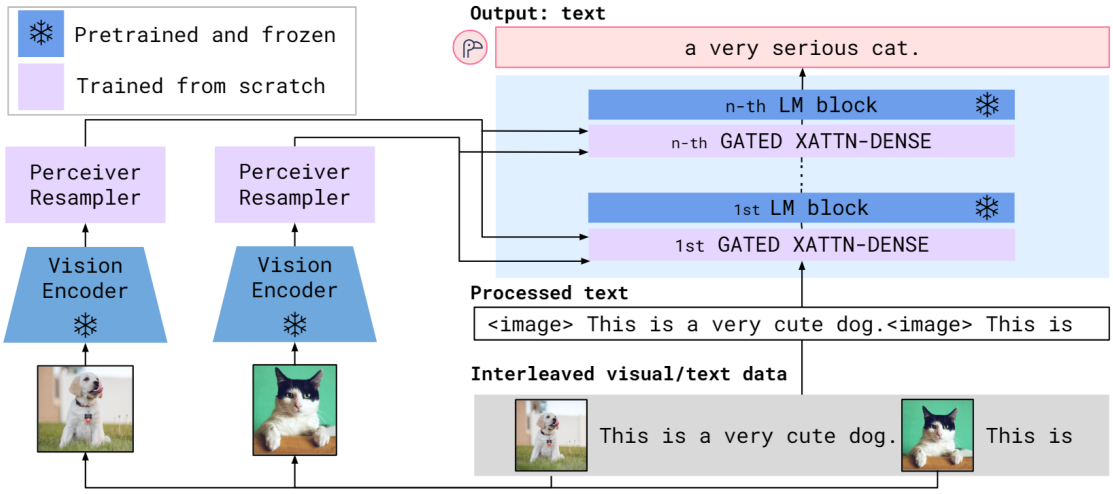


Fig. 2.15 Overview of the Flamingo [7] model architecture. This figure illustrates how Flamingo integrates visual and textual information through a combination of a Perceiver Resampler for visual data and gated cross-attention layers in the language model, enabling efficient few-shot learning across multi-modal tasks. The figure is from Alayrac et al. [7].

Flamingo (Fig. 2.15), built on Chinchilla [108], introduces an innovative model that integrates pre-trained vision and language models with new components for handling interleaved visual and textual data. It excels in few-shot learning, achieving strong performance on various benchmarks without task-specific fine-tuning. Key novelties include the Perceiver Resampler for converting visual features into tokens and gated cross-attention layers for conditioning on visual input, enabling rapid adaptation to new tasks with minimal data.

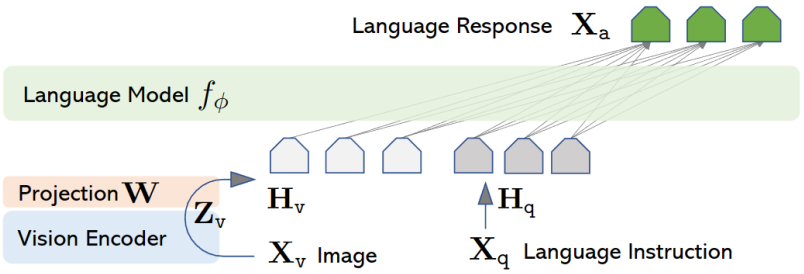


Fig. 2.16 Overview of LLaVA [198] architecture. The model integrates a vision encoder with a language model, fine-tuned on multi-modal instruction-following data generated by GPT-4. This combination enables advanced visual reasoning and instruction-following capabilities, setting new benchmarks in multi-modal AI performance. The figure is from Liu et al. [198].

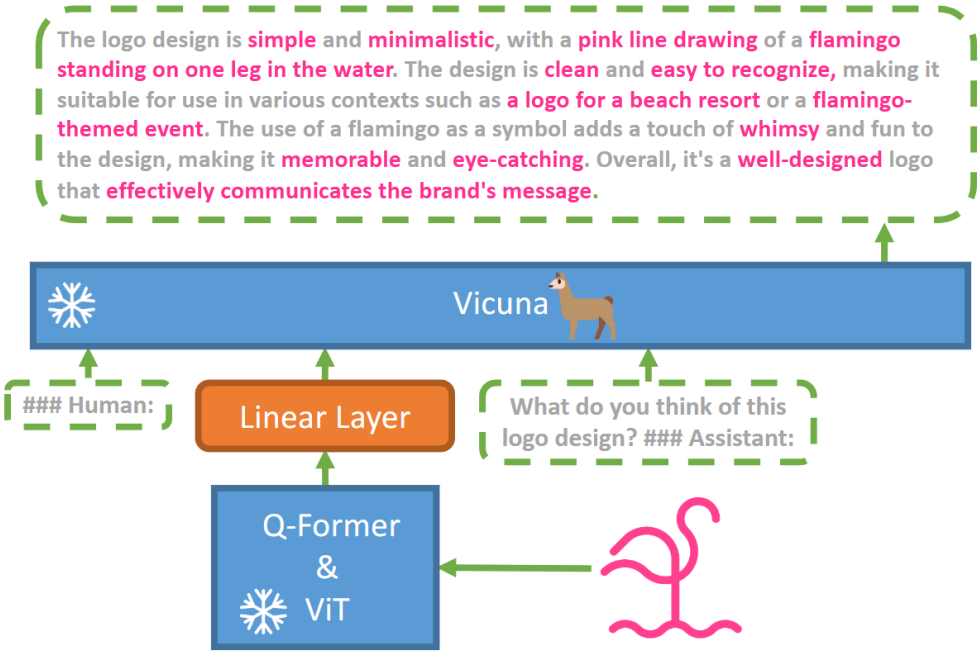


Fig. 2.17 The figure illustrates the architecture of MiniGPT-4 [388], highlighting the alignment between the frozen visual encoder and the language model via a linear projection layer. It demonstrates the model's capability to generate detailed image descriptions and perform complex vision-language tasks. The figure is from Zhu et al. [388].

LLaVA (Fig. 2.16), based on Vicuna [51], leverages conversational and interactive capabilities, making it effective for applications requiring dialogue and contextual understanding. MiniGPT-4 (Fig. 2.17), also built on Vicuna, focuses on training a lightweight mapping network to equip Vicuna with abilities to process multi-modal input, making it more accessible and efficient in resource-constrained environments.

InstructBLIP, combining FLAN-T5 and Vicuna, is built on BLIP2 (Fig. 2.11 and Fig. 2.12). The study uses 26 vision-language datasets, split into 13 for training and 13 for zero-shot evaluation. It emphasises instruction-following capabilities, suitable for multi-modal applications needing precise adherence to user instructions. It also improves generalisation to unseen tasks better than multitask learning and previous models like BLIP2 and Flamingo.

PALI-X, built on UL2 [299], is a multilingual vision and language model that excels in diverse and complex tasks across multiple languages. It achieves strong results in many vision-and-language benchmarks, demonstrating emergent capabilities such as complex counting and multilingual object detection. By scaling both vision and language components effectively, PaLI-X significantly improves over previous models, showcasing robust performance in multilingual settings.

QWen-VL (Fig. 2.14), based on QWen [14], inherits robust language understanding, making it powerful for multi-modal tasks requiring nuanced comprehension and generation. It supports multi-image inputs, multi-round dialogues, multilingual conversations, and fine-grained visual recognition.

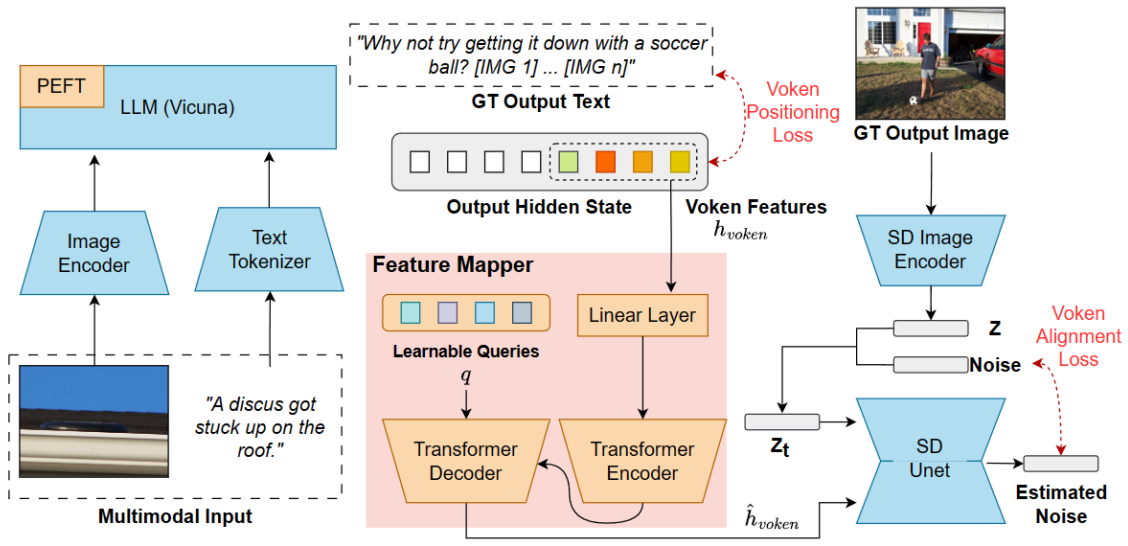


Fig. 2.18 Overview of MiniGPT-5 architecture. The figure demonstrates the interleaved vision-and-language generation process. The model integrates Stable Diffusion [257] and LLMs with generative vokens, enhancing multi-modal output coherence and quality. The figure is from Zheng et al. [380].

MiniGPT-5 (Fig. 2.18) introduces "generative vokens (i.e. visual tokens)" for seamless text-image integration, using a two-stage training strategy. The model employs Stable Diffusion [257] for image generation, achieving superior performance over previous models. This approach allows MiniGPT-5 to generate coherent multi-modal content without relying heavily on detailed image descriptions during training.

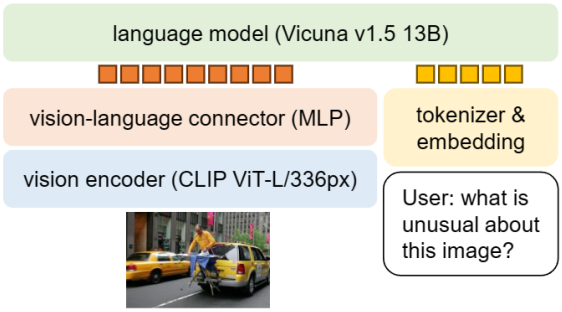


Fig. 2.19 The model architecture of LLaVA 1.5 [197]. The diagram illustrates the enhanced LLaVA model, integrating CLIP-ViT-L-336px with an MLP projection layer for improved vision-language alignment. The figure is from Liu et al. [197].

LLaVA 1.5 (Fig. 2.19), an improved version of LLaVA built on Vicuna 1.5 [51], is designed to enhance multi-modal models through several key innovations. A two-layer MLP projection is introduced to strengthen the vision-language connection, significantly enhancing performance. This model outperforms previous results on 11 benchmarks with only 1.2M data points, demonstrating exceptional data efficiency. Additionally, it is trained on a single 8-A100 node in one day, showcasing superior computational efficiency. It achieves enhanced performance using simpler architecture and less data compared to InstructBLIP and Qwen-VL.

MiniGPT-v2, derived from LLaMA 2 [304], is a vision-language model that excels in tasks like image captioning, visual question answering, and visual grounding. In contrast to the previous MiniGPT-4 model, its novelties include using task identifiers to enhance learning efficiency and reducing visual tokens by 75% through token concatenation. It concatenates 4 adjacent visual tokens in the embedding space and projects them together into one single embedding in the same feature space of the LLM, thus reducing the number of visual input tokens by 4 times, making it more capable of handling high-resolution images efficiently. The model undergoes a three-stage training process involving weakly-labelled, fine-grained, and multi-modal instruction datasets, outperforming previous models across various benchmarks.

These models illustrate the dynamic landscape of LMMs, showcasing how continuous advancements in both vision and language components drive significant improvements in multi-modal understanding and generation. The integration of foundational language models

with specialised techniques for processing visual data has led to a new era of AI capabilities. Each model contributes uniquely to this evolving field, whether through efficient learning mechanisms, multilingual support, or innovative training strategies. As these technologies progress, they pave the way for more sophisticated and versatile AI systems capable of seamlessly interacting with and interpreting diverse data modalities, ultimately broadening the horizons for practical applications in real-world scenarios. The synergy between foundational language models and advanced multi-modal models promises an exciting future for artificial intelligence, where the boundaries between different data types become increasingly blurred, allowing for more diverse ways of interacting with AI systems, such as uploading images, audio, and videos in interactions.

Approaches Extended from LLMs

Given the foundation of recent advanced LMMs on LLMs, similar methodologies can be adapted from LLM training to LMM training. For instance, LMMs can be fine-tuned using multi-modal instruction-following datasets to enable them to manage complex, instruction-aware tasks. For example, LLaVA, one of the powerful and popular LMMs, was trained on LLaVA-Instruct, a curated dataset introduced in the same paper. Chen et al. [31] presents ALLaVA, a lightweight LMM that is similar to LLaVA and was trained using synthetic data generated by GPT-4V (GPT-4 with Vision). By creating a high-quality dataset with detailed captions and reasoning instructions, ALLaVA achieves competitive performance on various benchmarks while being resource-efficient. This approach addresses the performance gap between large-scale and lightweight LMMs, highlighting the potential of synthetic data for training efficient models suitable for deployment on limited-resource devices. Zong et al. [393] introduces VLGuard, a curated dataset designed for safely fine-tuning LMMs to mitigate harmful content generation without sacrificing performance. Evaluations demonstrate that VLGuard significantly enhances safety, reducing adversarial attack success rates while maintaining helpfulness in various benchmarks.

In terms of leveraging the in-context learning capability of the backbone LLMs, Zong et al. [392] introduces VL-ICL Bench, a benchmark for evaluating multi-modal in-context learning in LMMs, addressing the limitations of current evaluations focused on VQA and image captioning. It assesses state-of-the-art models, revealing their strengths and weaknesses, and emphasises the need for future models to handle longer context lengths and better utilise examples.

Additionally, various alignment datasets tailored for multi-modal settings (e.g., LLaVA-RLHF [293], VLFeedback [172], and RLHF-V [360]) can be utilised to improve the models' ability to generate responses preferred by humans. For instance, Sun et al. [293] gathered

human preference data (emphasising greater helpfulness and reduced hallucination) from LLaVA, subsequently training the LLaVA-RLHF model. The proposed approach effectively reduces the hallucination issues that are common in LMMs, and the model tends to generate responses that are more favoured by humans, typically encompassing more reasoning and conveying greater amounts of information with regard to the input image. In these studies, the training techniques are generally the same as those utilised in LLM alignment training, including RLHF [54] and DPO [247].

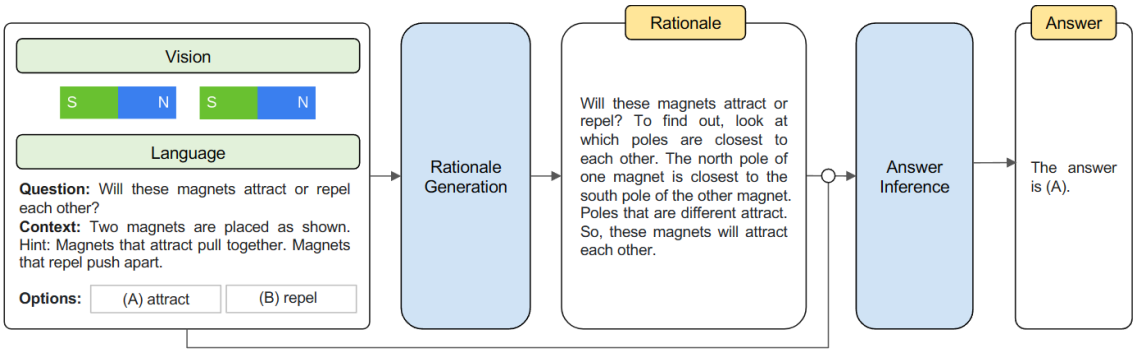


Fig. 2.20 Overview of Multi-modal Chain-of-Thought (CoT) prompting [373]. The diagram illustrates the reasoning process of LMMs. LMMs are able to read the figure and engage in sequential reasoning, thereby enhancing the accuracy of generated responses. The figure is from Zhang et al. [373].

Furthermore, leveraging the foundational LLMs, LMMs also incorporate the Chain-of-Thoughts mechanism, which breaks down tasks into steps to enhance the model's reasoning and problem-solving capabilities [373, 378, 276]. This mechanism is particularly effective in VL tasks where multi-modal understanding is critical. By decomposing complex tasks into manageable steps, LMMs can better interpret and generate contextually relevant responses. For instance, the Multi-modal CoT mechanism shown in Fig. 2.20 allows LMMs to process and analyse images sequentially, identifying key elements and their relationships before synthesising this information into a coherent narrative or description. This step-by-step approach improves the model's accuracy in tasks such as VQA, object recognition, scene understanding, and image captioning.

2.2 Information Retrieval

In this section, we present the principal advancements in Information Retrieval (IR) systems. We begin with text retrieval systems, which have been the subject of extensive investigation

in recent years, and subsequently address multi-modal retrieval systems, which are closely relevant to this thesis.

2.2.1 Text Retrieval

Retrieving relevant documents from databases based on user queries has been extensively investigated over the past several decades due to its importance in a wide range of applications, such as question answering systems, search engines, and recommender systems. Modern retrieval systems typically consist of two stages: Retrieval and Re-ranking [101]. These two stages often employ different models or systems with distinct focuses.

In the retrieval stage, a collection of coarse-grained relevant documents is retrieved from the databases. The primary focus of the model employed in this stage is the effective and efficient retrieval of documents from a large collection, optimising computational resources. In contrast, the model used in the second stage reorders the documents recalled in the first stage based on their finer-grained similarity to the query. Since ranking items more accurately according to the query is crucial, this stage typically utilises models that require more computational resources.

Conventional Retrieval Systems

Conventional text retrieval systems predominantly rely on term-based methods [101]. Term-based methods focus on matching query terms with document terms to identify relevant information. These methods typically involve techniques such as keyword matching, Boolean retrieval [271], and TF-IDF (Term Frequency-Inverse Document Frequency) [4, 255]. Keyword matching identifies documents containing the exact query terms. Boolean retrieval uses logical operators (AND, OR, NOT) to refine searches. TF-IDF assesses the importance of terms by evaluating their frequency within a document relative to their occurrence across the entire document collection. These approaches form the foundation for more advanced retrieval systems, providing a simple yet effective means of locating pertinent information.

Several approaches aim to enhance query and document representations by incorporating external resources or utilising the collection itself [241, 308, 367, 28, 2, 76]. Lexical dependency models, on the other hand, consider the relationships and order of terms to capture the meaning of texts more effectively [80, 261].

Another significant approach is the use of topic models [159, 64, 323, 67, 20], which have been extensively applied, particularly in 2016, due to their efficacy in capturing semantic relationships between words and identifying the topics within texts. Topic models train on textual data and represent queries and documents as vectors, with each dimension corre-

sponding to a specific topic. The most notable models in this category are Latent Semantic Indexing (LSI) [64] and Latent Dirichlet Allocation (LDA) [20].

LSI is a mathematical technique employed in natural language processing and information retrieval to analyse and uncover the underlying structure of text corpora. It represents documents and terms in a high-dimensional space, capturing relationships based on their co-occurrence patterns. By applying Singular Value Decomposition (SVD) [92], LSI reduces the dimensionality of this space, thereby revealing latent semantic relationships between terms and documents. This process enhances the retrieval and analysis of textual data by capturing the intrinsic semantic meaning beyond simple keyword matching.

LDA is a probabilistic generative model widely used for topic modelling. It posits that documents are represented as random mixtures over latent topics, with each topic being characterised by a distribution of words. LDA aims to uncover these latent topics by iteratively assigning words in documents to topics and inferring the underlying topic structure. The model employs the Dirichlet distribution to model the topic distributions and the multinomial distribution to model the word distributions within topics.

Sparse Retrieval Systems

Sparse retrieval methods are designed to enhance the efficiency of processing large document collections. These methods represent documents and queries using sparse vectors, which activate only a limited number of dimensions. They utilise term weighting schemes, such as TF-IDF and BM25 (Best Matching 25) [256], to efficiently associate documents with queries. TF-IDF assigns weights to terms based on their frequency within a document and their rarity across the entire collection, prioritising terms that are more discriminative and informative for retrieval. BM25, an improvement over the traditional TF-IDF model, is a ranking function used in information retrieval to estimate the relevance of documents to a given search query. It considers factors such as term frequency and document length to score documents.

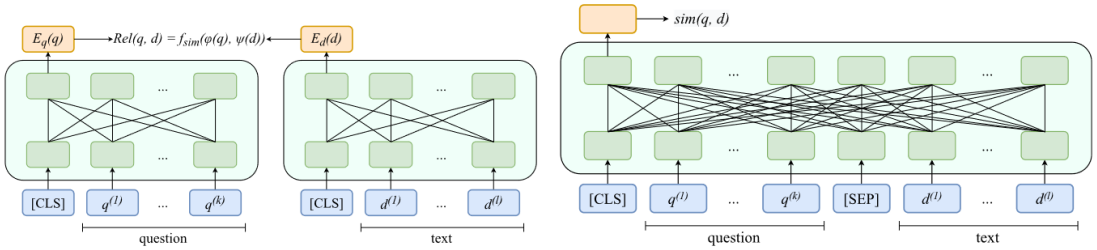
Recent research has employed neural models to enhance term weighting schemes while preserving the symbolic encoding of queries and documents [379, 394, 63]. Another line of work directly extracts latent sparse vectors using neural models [366, 337].

Dense Retrieval Systems

Deep learning models has significantly revolutionised this field by introducing neural networks capable of extracting dense and discriminative features for both queries and documents. In contrast to traditional sparse retrieval models, which depend on sparse representations

(e.g., bag-of-words [102]), dense retrieval models utilise dense vector representations to encapsulate semantic similarity between queries and documents. The superior text representation capabilities of pre-trained language models (such as BERT) and the availability of large-scale labelled datasets (such as MSMARCO [16] and Natural Questions [156]) have both contributed to the rapid advancement of dense retrieval models.

One of the primary advantages of dense retrieval models is their capacity to capture nuanced semantic relationships between words and phrases, thereby improving the relevance of retrieved documents in comparison to traditional sparse retrieval methods. Additionally, dense representations facilitate more flexible and scalable indexing strategies, enabling faster retrieval across extensive text collections.



(a) Dual-encoder architecture (b) Cross-encoder architecture

Fig. 2.21 Illustration of bi-encoder (left) and cross-encoder (right) models. The figure is from Zhao et al. [376].

The cross-encoder architecture (Fig.2.21 right) encodes queries and documents together, exhaustively for all query-document pairs. This approach achieves high precision; however, it is impractical for real-world deployment when the document database reaches a scale of billions. Conversely, the bi-encoder architecture is widely utilised in the development of dense retrieval models (Fig.2.21 left). A notable example of a bi-encoder (or dual-encoder) dense retrieval model is Dense Passage Retrieval (DPR) [143], which has been the subject of extensive research over the past few years. In this architecture, a query encoder processes the query, while a context encoder processes the documents, both converting them into one-dimensional dense vector representations using the '[CLS]' token[1] of the base BERT model. The similarity score between the query and document is then computed via the dot product of their respective vector representations. Retrieval latency is significantly reduced through the use of vector databases, such as FAISS [138], which support fast Approximate Nearest Neighbour Search (ANNS) [19, 100, 174]. Due to their robust retrieval capabilities and high efficiency, models with a similar architecture to DPR have been extensively studied

[1]'[CLS]' token is a pre-defined first token at the input of many encoder models including BERT, and its representations are often used in classification tasks.

in subsequent research [243, 330, 229, 230]. For instance, recent studies have employed T5 as the query and document encoders, achieving strong performance across a range of retrieval tasks [229, 230].

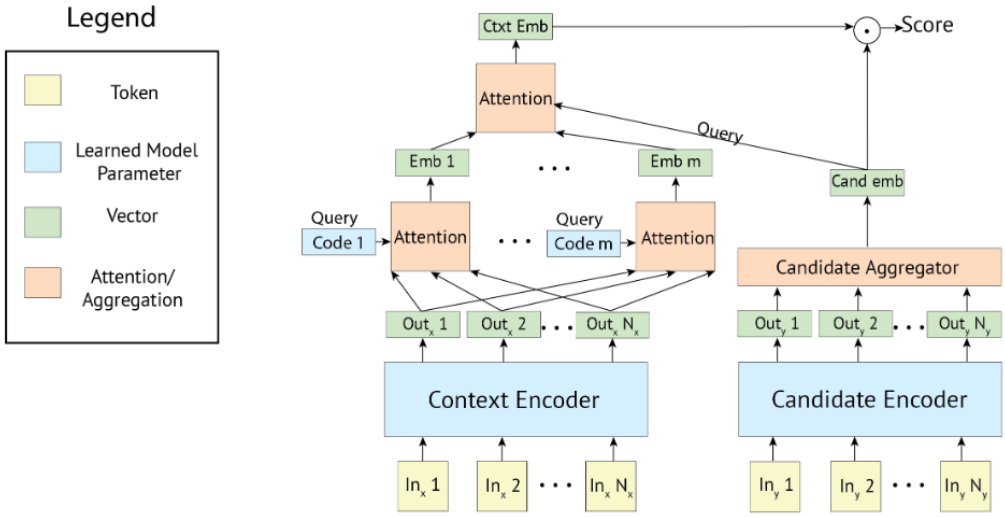


Fig. 2.22 The figure illustrates the Poly-encoder [121] architecture, combining global attention features with self-attention mechanisms. It balances the computational efficiency of bi-encoders and the high accuracy of cross-encoders. The architecture uses multiple context codes and attention heads to effectively score the similarity between contexts and candidates, optimising both speed and performance. The figure is from Humeau et al. [121].

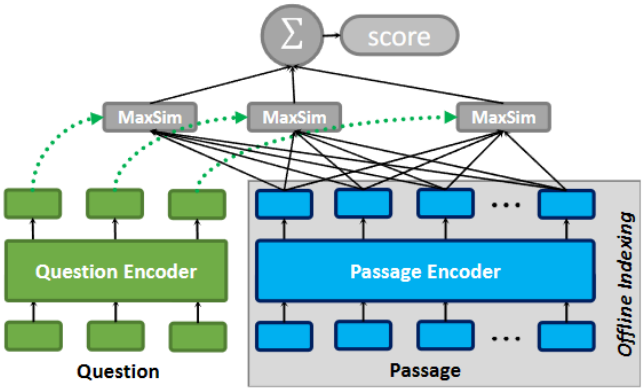


Fig. 2.23 Illustration of late-interaction models (ColBERT). The query features and document features interact with each other at the token level, making it capable of capturing fine-grained relevance. The figure is from Santhanam et al. [264].

A significant limitation of one-dimensional bi-encoder representations lies in their restricted ability to model fine-grained semantic interactions between queries and texts. Another research direction aims to extend query/document representations to multi-dimensional

vectors [121, 207, 146, 109, 264, 160]. Humeau et al. [121] introduced the Poly-encoder (Fig. 2.22), which employs multiple *context codes* (trainable parameters) that are aggregated into the query embeddings via attention mechanisms during retrieval. The Poly-encoder substantially outperforms bi-encoder models and achieves performance comparable to cross-encoder models, while maintaining retrieval latency at a similar scale to bi-encoder models, as reported in Tables 4 and 5 in the original paper. Furthermore, late-interaction models, such as ColBERT [146], ColBERTer [109], and ColBERTv2 [264], utilise entire token-level representations from queries and documents (Fig. 2.23). The final relevance score is computed by aggregating the token-level similarity scores between every pair of query and document tokens (as shown in Fig. 2.23), which provides a more fine-grained approach and significantly improves retrieval performance compared to one-dimensional DPR models. The expansion of this approach has been investigated in subsequent studies [56, 129, 83, 228, 32], resulting in enhanced performance across a variety of tasks. The introduction of the PLAID engine [263] has significantly advanced this line of work by greatly reducing the computational requirements of late-interaction retrieval.

In the training of dense retrieval models, contrastive loss [143] (or alternative loss functions such as triplet loss [268]) is utilised to shape the latent embedding space 'conceptually'. In the latent space, relevant documents are positioned closer to the query, while irrelevant documents are positioned further away. The training data typically comprises a query, a relevant document (positive), and several irrelevant documents (negative). Negative sampling is a common technique used to identify these negative examples by sampling documents from the remaining documents (excluding the positive documents).

Dense retrieval models have been applied to various NLP tasks, including information retrieval, question answering, document summarisation, and conversational systems, among others. Their efficacy in capturing semantic similarity and their ability to scale to large datasets make them a promising approach for addressing the challenges of information retrieval in modern data-driven applications.

2.2.2 Cross-modal Retrieval

The preceding subsection examined fundamental retrieval models within text-only contexts. From this subsection onwards, we will explore retrieval methodologies that extend beyond text-only queries and documents.

In recent years, there has been significant research in the field of cross-modal information retrieval, with particular emphasis on aligning the latent representations of images and text. Transformer models have become essential tools in this area, and are extensively employed to process multi-modal data.

The evaluation of these models frequently utilises benchmark datasets such as MSCOCO [142], Flickr30k [239], WIT [289], and IGLUE [24]. These models are tasked with either retrieving the textual description that corresponds to a given image (Image-to-Text Retrieval) or identifying the image that matches a provided textual description (Text-to-Image Retrieval).

By aligning the representations of visual and textual modalities within these models, there is a notable enhancement in their ability to interpret both complex images and textual content. Consequently, incorporating image-text matching tasks into the pre-training phase of large Vision-Language (VL) models, such as BLIP2, has become a standard practice.

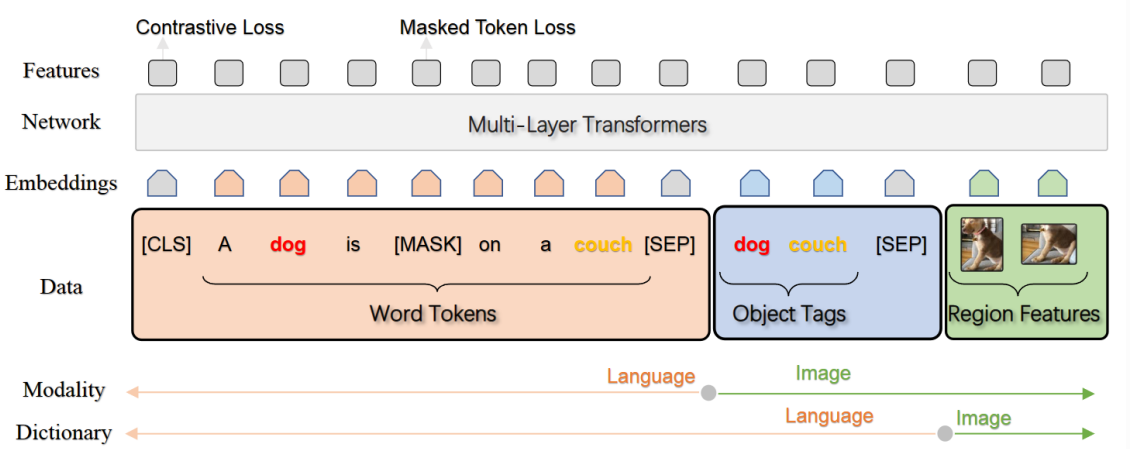


Fig. 2.24 The model architecture of Oscar [175]. A detailed explanation of this system can be found in the main content. The figure is from Li et al. [175].

Many cross-modal retrieval models utilise object detectors to identify regions of interest within an image and extract the corresponding regional features. For example, during the pre-training phase of Oscar [175] (as illustrated in Fig. 2.24), an object detector is used to recognise objects in an image. The textual labels of these detected objects, along with their regional features, are then input into Transformer layers. This integration of object detection enhances the alignment of object-level semantics with textual representations. To train the model's language understanding capability and refine this alignment, Mask Token Loss and Contrastive Loss are employed: Mask Token Loss trains the model to predict randomly masked tokens based on both the input image and text, thereby enhancing the joint understanding of image and text, while Contrastive Loss involves randomly replacing the input object tags with a different tag sequence sampled from the dataset. It then trains the model to determine whether the provided tags are consistent with the input image, thereby increasing the model's awareness of the relevance between text and image. Similarly, VILLA [87] introduces an innovative approach to large-scale adversarial training for vision-and-language representation learning. It also integrates regional features obtained through

object detectors to enrich image representations. By conducting adversarial training within the embedding space of each modality, VILLA achieves notable performance improvements in various downstream tasks such as VQA.

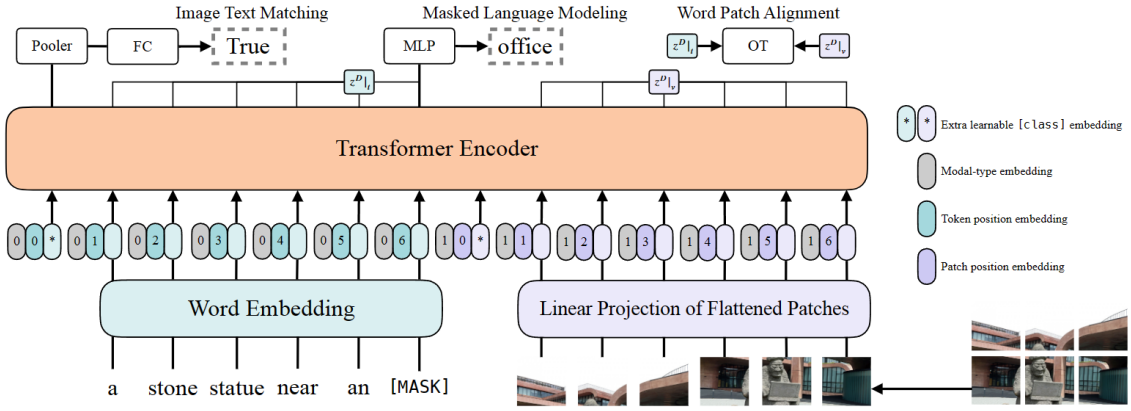


Fig. 2.25 The model architecture of ViLT [148]. ViLT segments images into patches and directly inputs them into Transformer layers. The figure is from Kim et al. [148].

Recently, there has been a growing interest in developing multi-modal models capable of handling visual modality without the need for explicit object detectors for region-level feature extraction. ViLT (Vision and Language Transformer) [148] (Fig. 2.25) segments images into patches and directly inputs them into Transformer layers. ViLT has achieved comparable, and even stronger on some benchmarks, results when compared to models with object detectors. The model is trained with three key objectives: (1) Image-Text Matching (ITM): The model learns to predict whether a given image and text pair are related or not. This helps the model understand the relationship between visual and textual information. (2) Masked Language Modeling (MLM): The model predicts missing words in a sentence, enhancing its language understanding by leveraging contextual clues from both text and image. (3) Whole Word Masking: Instead of masking random subwords, entire words are masked to make the prediction task more challenging and realistic, improving the model's linguistic capabilities. The proposed approach has led to significant advancements, particularly in cross-modal retrieval performance. The efficiency of both training and inference processes has seen substantial improvements due to the elimination of explicit object detectors. SimVLM [318] follows a similar approach by utilising a visual encoder to embed images, highlighting the dispensability of explicit object detection in the construction of Vision and Language models.

Another significant research direction involves encoding image and text data into global features separately [246, 130, 346]. CLIP (Contrastive Language-Image Pretraining) [246] employs a contrastive loss (discussed in Sec. 2.2.1) in training, directly aligning image embeddings with text embeddings (Fig. 2.26). This method utilises distinct visual and text

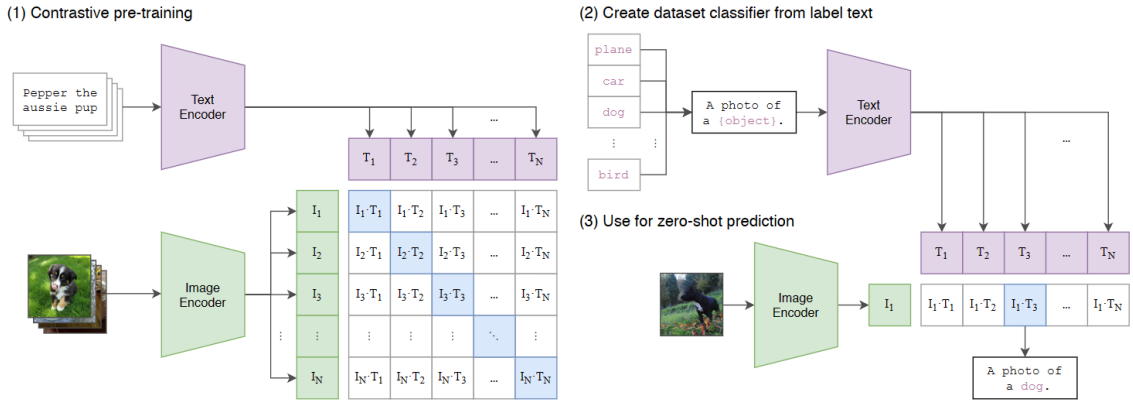


Fig. 2.26 The illustration of CLIP (Contrastive Language-Image Pretraining) [246]. CLIP encodes images and texts separately, and than trains the model with contrastive learning. The text that describes the image well will be assigned a higher relevance score. The figure is from Radford et al. [246].

encoders to independently encode image and text data like a bi-encoder (Sec. 2.2.1), which enables rapid retrieval with vector databases that support Approximate Nearest Neighbour Search.

This technique is considerably more efficient than traditional methods, such as calculating relevance scores for each image-text pair in the corpus exhaustively, as exemplified by models like Oscar. Despite relying solely on large-scale pre-training data, CLIP ViT models (e.g., CLIP ViT-B/L) and its subsequent scaled-up versions (e.g., OpenCLIP ViT-g/H released by Cherti et al. [50]) achieve exceptional results. These models are notable for their remarkable performance and are widely used as the foundational visual encoders for developing state-of-the-art vision-language models. For instance, many LMM models, such as LLaVA 1.5 (illustrated in Fig. 2.19), employ CLIP's visual encoder as the core vision model; Stable Diffusion [257] leverages CLIP's text encoder as the foundational text model, effectively guiding the diffusion process with prompts. Building on the architecture of CLIP, ALIGN [130] demonstrates the efficacy of training models on a dataset that captures the natural distribution of image-text pairs. With the introduction of this new dataset, ALIGN achieves robust visual and vision-language representations.

CLIP-like models encode both images and texts into single vector representations, which may constrain their ability to capture detailed information in cross-modal interactions. FILIP [346] addresses this limitation by enabling cross-modal late interaction, thereby enhancing the model's capacity to capture token-level information between images and texts, as illustrated in Fig. 2.27.

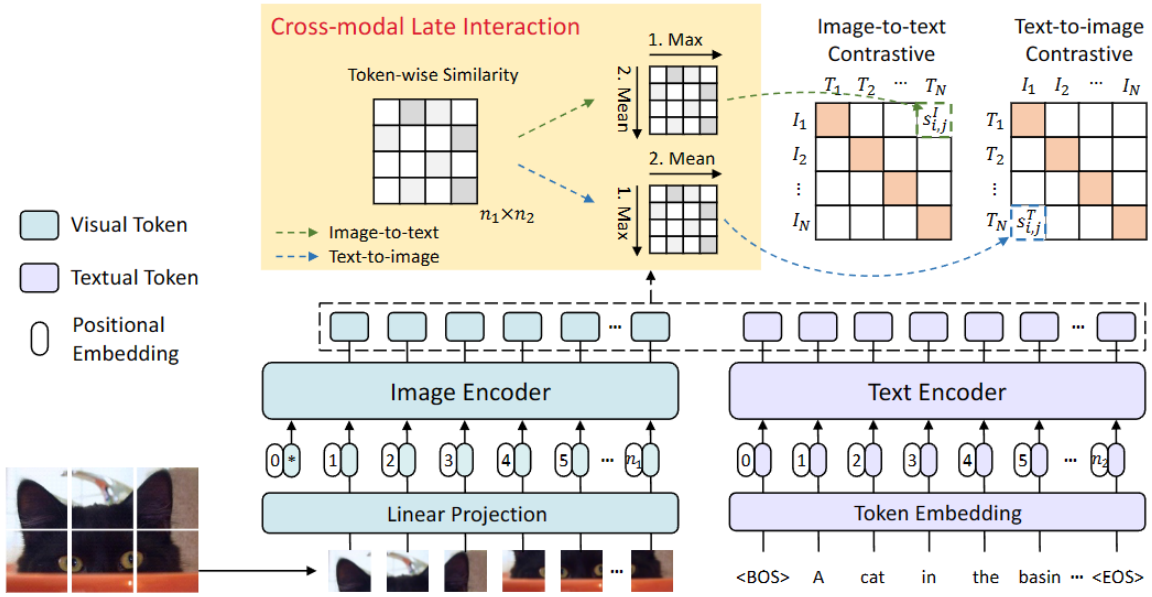


Fig. 2.27 The model architecture of FILIP [346]. The late-interaction design captures the fine-grained image-text interactions. The figure is from Yao et al. [346].

Some hybrid approaches [169, 356, 17, 170, 312] implement separate encoding processes for images and texts, subsequently utilising the derived features within a multi-modal encoder. For instance, as depicted in Fig. 2.28, ALBEF [169] initially encodes images and text separately, and the resultant features are input to a multi-modal encoder for integration. ALBEF employs contrastive learning to enable the model to achieve semantically aligned representations of the modalities, thus enhancing the multi-modal learning process of the fusion model. Nonetheless, this type of models necessitates per-instance evaluation, a requirement that may present significant challenges, particularly when performing image-text matching on a large-scale dataset.

2.2.3 Multi-modal Retrieval

Cross-modal retrieval seeks to match images with text, whereas recent advancements in multi-modal information retrieval focus on finding relevant documents using multi-modal queries comprising both images and text. This is particularly beneficial for retrieving documents to answer knowledge-intensive VQA queries.

Given the extensive volume of documents in knowledge bases, the bi-encoder architecture is predominant in this domain. This architecture consists of a query encoder that encodes the multi-modal query and a context/document encoder that encodes (multi-modal) documents.

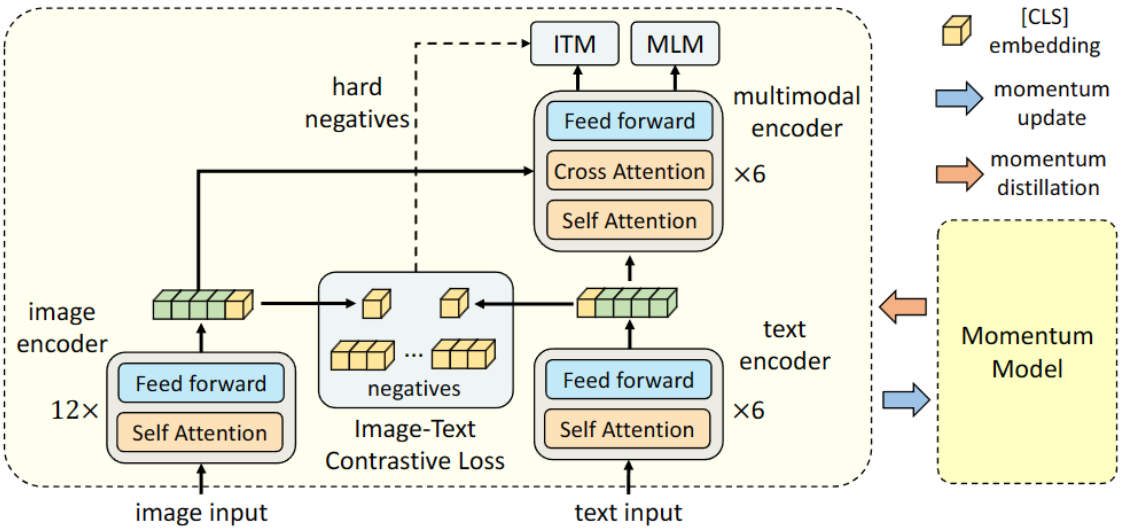


Fig. 2.28 The model architecture of ALBEF [169]. It initially encodes images and text separately, and the resultant features are input to a multi-modal encoder for integration. The figure is from Li et al. [169].

Some early works [88, 183, 208, 114] convert images into textual form, enabling direct processing by a text-only query encoder. For instance, Lin and Byrne [183] (our work, introduced in Chapter 3) extract image captions from a captioning model, object tags detected by an object detector, and OCR (Optical Character Recognition) detection results. These textual descriptions of images are used to extract relevant documents using Dense Passage Retrieval (DPR).

In contrast, recent studies employ multi-modal Transformers to jointly encode images and text queries into compact, single-dimensional embeddings. Luo et al. [208] utilise LXMERT [294] as the query encoder to process both the image and the question. Lerner et al. [164] trains the retriever with a multi-modal inverse cloze task, significantly improving retrieval performance on ViQuAE [163]. Specifically, the sentence within a Wikipedia passage is treated as a pseudo-question, while the rest of the passage provides the context as a pseudo-relevant passage for training. Additionally, the image accompanying this sentence is included in the pseudo-question while the image in the infobox of this Wikipedia page is included as a part of the pseudo-relevant passage; a follow-up work [165] encodes the multi-modal query by a weighted sum of BERT [66] and CLIP [246] embeddings. It shows that by combining mono- and cross-modal retrieval in training, the approach achieves better performance and efficiency compared to solely using mono-modal or more complex, larger models. More recently, Zhou et al. [384] introduces a method that integrates a visual encoder with a pre-trained text retriever, resulting in a unified multi-modal representation. This approach involves generating high-quality image-text pairs and utilising a multi-stage

training process. Initially, the visual token embedding is aligned with the text encoder using a substantial amount of weakly labelled data. Subsequently, the method enhances multi-modal representation capabilities through the use of the generated composite image-text data.

Apart from these single-dimensional embedding approaches, FLMR (Fine-grained Late-interaction Multi-modal Retriever) [188] (our work, introduced in Chapter 5) introduces an innovative retriever designed specifically for multi-modal retrieval in knowledge-intensive VQA tasks. It captures fine-grained visual information in images by leveraging token-level interactions and enriching the query with regional features derived from object detection. PreFLMR (our work, introduced in Chapter 6) further examines the scalability of FLMR. When scaled up to billions of parameters and millions of training data pairs, PreFLMR demonstrates robust multi-modal retrieval performance across nine different datasets.

2.3 Retrieval Augmented Generation

In this section, we provide a brief overview of notable advancements in the development of RAG systems. We commence by highlighting the progress made in text-only RAG systems before moving on to discussions concerning multi-modal settings. Specifically, we direct attention towards vision and language RAG, as it is most close to the scope of our research.

2.3.1 Text-only RAG

RAG is a method that combines traditional text generation with retrieval-based techniques. It involves retrieving relevant information from a database or corpus to enhance the generated output, ensuring it's more accurate, contextually relevant, and diverse. It is also called "retriever-reader" architecture in the literature. We will also use this term in some of the chapters. With retrieval methods already introduced in earlier sections, here we focus on the popular approaches to leverage the retrieval models in content generation.

Leveraging Retrieved Content in Generation

Augmenting queries with explicitly retrieved content is a widely employed approach in numerous recent RAG systems [249, 99, 167, 12, 284, 202, 345, 350, 206, 137]. This method involves directly concatenating the user's query with the retrieved content and sending the enriched query to the generator, which is typically an LLM capable of reasoning and generating responses using the retrieved content.

For instance, REALM [99] and RAG [167] are two landmark models that transmit concatenated queries to the generator to produce more reliable and accurate responses for

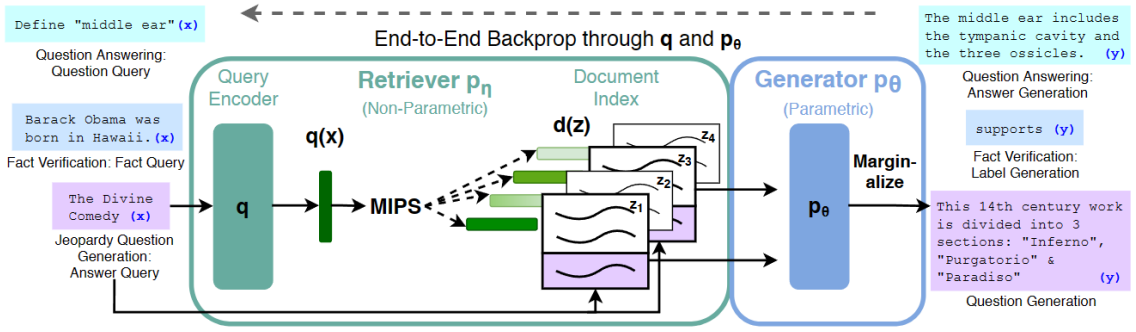


Fig. 2.29 The RAG model proposed by Meta [167]. The generation probability y is marginalised over all retrieved documents $(z_1, z_2, ...)$: $p(y|x) \approx \sum_{k=1}^{K} p_\eta(z_k|x)\ p_\theta(y|x, z_k)$, where p_η is the retriever and p_θ is the generator in the figure; $p_\eta(z|x)$ is the retrieval probability over all K retrieved documents computed using $q(x)$ and $d(z)$, the dense vectors of query x and document z, respectively. Then the loss is back-propagated through both the generator and the retriever to enable joint optimisation. The figure is from Lewis et al. [167].

tasks such as question answering. Notably, these models jointly optimise the retriever during the training of the generator by leveraging a loss function that marginalises the joint probability over the retrieved documents (as illustrated in Fig. 2.29). SELF-RAG [12] utilises a specialised critique model to ascertain whether retrieval is necessary to produce an accurate response. REPLUG [284] conducts retrieval offline and then exploits the pre-trained reasoning capabilities of LLMs via APIs. Additionally, some code completion models follow a similar approach [206, 137].

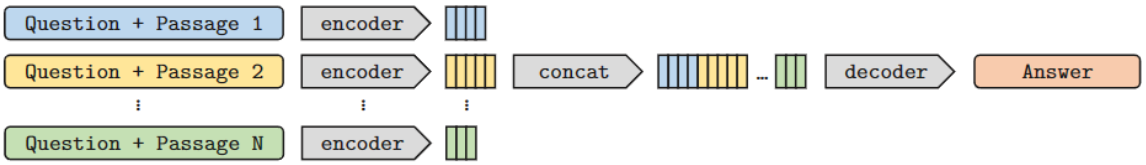
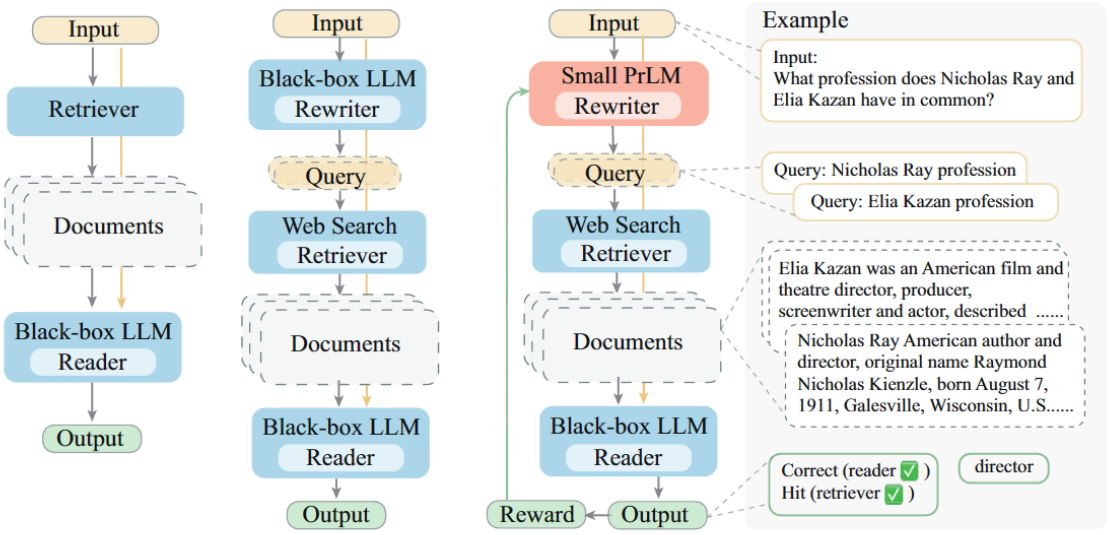


Fig. 2.30 The illustration of FiD (Fusion-in–Decoder) [124]. The outputs of the encoder are concatenated and fed to the decoder to generate the response. The figure is from Izacard and Grave [124].

Another line of research [21, 124, 355, 283] transforms retrieved content into latent representations, which are subsequently incorporated into the generator model as embeddings. For instance, as depicted in Fig. 2.30, the FiD (Fusion in Decoder) model [124] utilises an encoder to process the query along with each retrieved document, concatenating the encoder outputs. A separate decoder model then generates an answer by cross-attending to these concatenated latent representations. The FiD architecture has been widely adopted in numerous recent RAG systems [88, 355, 283].

In addition to the aforementioned approaches, some models integrate the retriever logits during generation [145, 103, 382, 222], for example, by aggregating the retrieval confidence/probabilities of each retrieved document and using them in the generation process.

Query Rewriting



(a) Retrieve-then-read (b)Rewrite-retrieve-read (c) Trainable rewrite-retrieve-read

Fig. 2.31 The pipeline introduced by Ma et al. [210] (right). A rewrite model is trained with reinforcement learning to effectively tailor the input query for retrieval. The figure is from Ma et al. [210].

Recent research has introduced many methodologies aimed at enhancing overall system performance. Among these, a commonly employed technique involves transforming input queries to render them more informative and suitable for retrieval. This process entails rewriting the query using specialised models or LLMs, thereby refining the representation of user intent and improving alignment with retriever models. For instance, Ma et al. [210] (as illustrated in Fig. 2.31) employ reinforcement learning to train a rewriter model, thus enhancing query rewriting and subsequently improving retrieval performance. Similarly, TOC [147] utilises retrieved content to decompose ambiguous queries into distinct sub-queries, which are then processed by a generator and aggregated to yield the final outcome. Additionally, Query2doc [310] and HyDE [89] adopt the original query to generate a pseudo-document, subsequently employed as the retrieval query. This pseudo-document encapsulates rich, relevant information, facilitating the retrieval of more precise results. Query rewriting is also effective in conversational search [213], where it extracts and understands user search intent within conversational contexts.

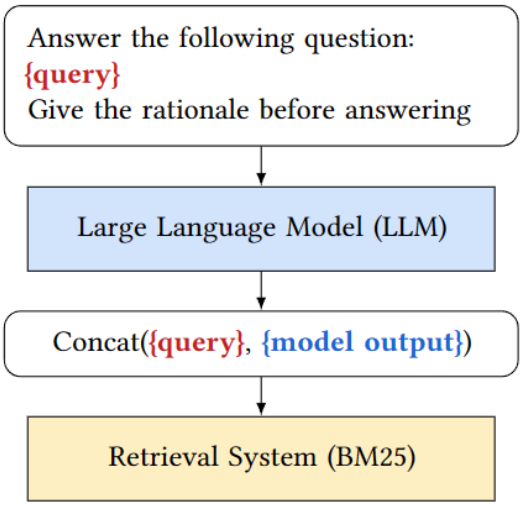


Fig. 2.32 Query expansion with Chain-of-Thought prompting introduced by Jagerman et al. [125]. The figure illustrates the flow from the initial query through various prompting techniques, including zero-shot, few-shot, and Chain-of-Thought, ultimately leading to the expanded query terms. It has been demonstrated that Chain-of-Thought prompting significantly enhances retrieval performance. The figure is from Jagerman et al. [125].

With appropriate prompting techniques, LLMs can proficiently augment input queries. Recent studies show that query rewriting can be effectively accomplished through various prompting methods, including zero-shot prompting [125, 280, 6, 85, 211, 361], few-shot prompting [310, 125, 6, 361], and Chain-of-Thoughts (CoT) prompting [6, 125].[2] For example, Jagerman et al. [125] explores using LLMs for query expansion to enhance search recall. As depicted in Fig. 2.32, the LLMs are prompted to generate responses with their internal knowledge, which serve as additional relevant search terms. The LLMs provide more accurate and contextually appropriate expansions compared to traditional methods. They investigate different prompting techniques, including zero-shot, few-shot, and CoT prompts, with CoT significantly improving relevance.

To achieve superior rewriting performance tailored to specific downstream applications, LLMs can be fine-tuned for query rewriting [238, 210]. For example, Peng et al. [238] utilise LLMs to expand, rectify, and enhance user input queries, thereby improving the accuracy of e-commerce product searches. To mitigate latency associated with LLMs in rewriting, Srinivasan et al. [291] distil the query rewriting capability of the LLM into a smaller specialised model, which exhibits faster inference and enhanced scalability in deployment.

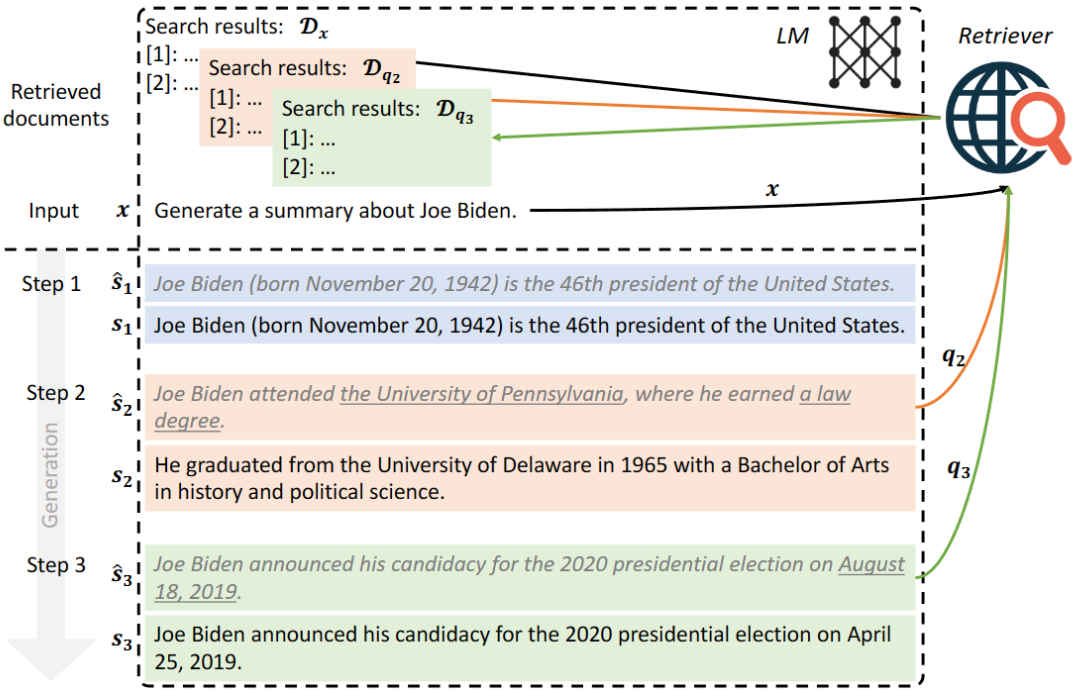


Fig. 2.33 The FLARE (Forward-Looking Active REtrieval augmented generation) pipeline [136]. The model actively calls the retriever if the candidate generation contains low-confidence tokens. The figure is from Jiang et al. [136].

Adaptive Retrieval

An additional enhancement to the original RAG pipeline is the incorporation of adaptive retrieval mechanisms. Retrieval processes may not always be necessary, particularly when the internal knowledge of LLMs is adequate for addressing relatively straightforward inquiries. The integration of retrieved content could potentially result in resource inefficiencies and the introduction of confusion in model generation [375].

The decision regarding whether to initiate a retrieval operation can be guided by predefined rules [103, 136, 212, 134, 140] or by models [12, 252, 314, 68, 127]. For example, as shown in Fig. 2.34, FLARE (Forward-Looking Active REtrieval augmented generation) [136] employs a two-step process wherein it initially generates a candidate sentence and subsequently invokes the retriever if certain tokens within the generated content exhibit low confidence scores. In contrast, AdaptiveRAG [127] utilises a smaller LM to evaluate the complexity of the input query and triggers the retriever as deemed necessary.

[2]Introduced and discussed in Sec. 2.1.3.

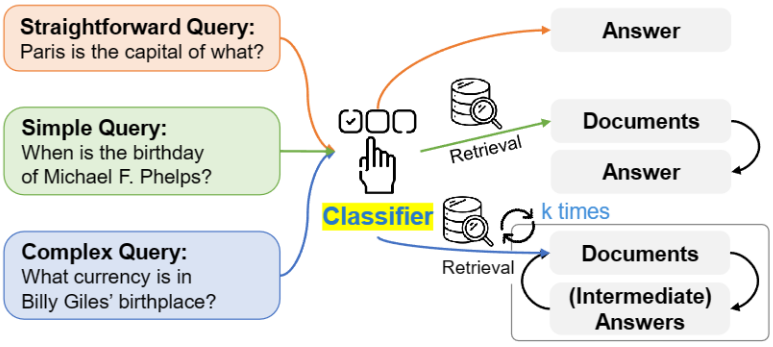


Fig. 2.34 The pipeline of AdaptiveRAG [127]. A classifier is employed to assess the complexity of the input query. Then the system calls the retriever or starts the generation without retrieval. The figure is from Jeong et al. [127].

2.3.2 Vision-and-Language RAG

In this subsection, we briefly summarise some research work that leverage RAG to solve vision-and-language tasks, with most of them further discussed in detail in Sec. 2.4.1. Notably, many techniques described in the preceding subsection (Sec.2.3.1) are applicable to both Text-only RAG and Vision-and-Language RAG.

In open-domain VQA, numerous studies utilise RAG to produce answers grounded in documents retrieved from external knowledge sources. Notable examples include RA-VQA [183], RA-VQA-v2 [188, 190], VisualDPR [208], TRiG [88], MuRAG [36], and RAMM [365]. LLMs serve as an internal knowledge source to provide answer candidates in works such as PICa [341] and KAT [96].

Beyond VQA, which is the primary focus of this thesis, vision-and-language RAG has also been explored in image captioning [386, 52, 265, 285, 250, 343, 385], visual dialogue systems [81, 177, 279], and text generation [340, 389, 82, 37, 349]. Due to the limited scope of this thesis, these areas will not be discussed further.

2.4 Visual Question Answering

In this section, we focus on describing one of the two major tasks examined in this thesis: Visual Question Answering (VQA).

We begin with an overview of Vision-and-Language tasks (Sec. 2.4.1) to provide a broader context for the challenges in understanding both vision and language, before moving on to a detailed discussion of VQA tasks. Sec. 2.4.2 introduces widely-used VQA/KB-VQA datasets, followed by two sections that discuss recent advancements in VQA systems (Sec. 2.4.3) and

KB-VQA systems (Sec. 2.4.4), respectively. The research challenges in developing KB-VQA systems are analysed to contextualise the study presented in this thesis.

2.4.1 Overview of Vision and Language Tasks

With different task objectives, vision-and-language (VL) tasks can be assigned to one of several categories:

1. **Visual Question Answering (VQA):** given an image, a question is posed to explore the relationships (including spatial relation (e.g. up, below, and on) and semantic relation (e.g. holding and wearing)) between visual elements in the image. An example question is "Q: what is the man wearing? A: sunglasses". In "*Knowledge-based* VQA" (KB-VQA), answering a question requires external knowledge and potentially commonsense reasoning ability. Some comonsense/other knowledge bases (e.g. ConceptNet [288] and Wikipedia) are widely employed for retrieving relevant knowledge to help answer questions [311, 214].

2. **Visual Dialogue System:** given an image, an automated agent engages a human in a meaningful multi-turn conversation. Specifically, given an image, a dialog history, and a follow-up question about the image sent by the human which are often tightly related to the previous dialogue context, the system's task is to answer the follow-up question. A more recent and more difficult task brings in multi-modal dialogue input: images related to the conversation change dynamically, e.g., in automated customer service where a user sends more images as the dialogue progresses [374].

3. **Vision-and-Language Navigation (VLN):** Given general, verbal instructions, the system attempts to complete the required tasks, such as moving in space and fetching items. Relevant high-quality data in this field is relatively rare. Two of the popular datasets are Room2Room (R2R) [9] and REVERIE [240]. The data is normally collected from software simulation platforms. For example, Matterport3D was introduced in [240] to simulate an embodied agent. The software contains 3D viewpoints distributed throughout the entire walkable floorplan of each scene of a house at an average separation of 2.25m. Each view is comprised of 18 images captured from a single 3D position at the approximate height of a standing person. Finally, human annotators used the simulator software to annotate verbal instructions and corresponding simulated walk paths, which were then used as labelled data for VLN tasks. An example of a difficult VLN task is "move to the bedroom and tell me what is the color of the wall",

which requires environment sensing (find the door to the bedroom), robot control, and question answering.

4. **Captioning:** Given an image or a video clip, the system generates a short text that identifies the visual elements and describes their relationships, such as "a cat sits in front of a TV". A well-known dataset is Microsoft COCO [181], where each image comes with 5 associated captions. The authors of the Visual Genome dataset leveraged manually-designed rules to generate a huge amount of captions for each image from dense annotations of objects and their relationships [153].

5. **Storytelling:** From images or videos, the system produces written stories in natural language. Automated storytelling can be used for writing headlines, financial reports and weather updates, screenplays, and short stories [119, 331, 195, 194].

Among these VL tasks, this thesis focuses on VQA, which was considered the most promising direction for long-term research at the beginning of this Ph.D. program. The reasons are:

(1) VQA establishes a foundation for Visually-grounded Language Systems by enabling models to comprehend multi-modal information in the input. Investing in enhancing this capability paves the way for future advancements in other VL tasks, such as Vision-and-Language Navigation;

(2) There is a growing body of work dedicated to the creation of new VQA datasets and the updating of existing ones;

(3) Knowledge-based question answering represents a highly promising research subfield. It is essential for addressing more complex queries posed by real users. These potential queries often pertain to real-world knowledge, such as "How old is that actor?" and "Is that first restaurant any good?". Consequently, Vision-and-Language systems must be capable of understanding these questions and providing answers by retrieving external knowledge when necessary.

Currently, there is a significant demand for large multi-modal systems that can answer multi-modal, knowledge-intensive questions.

In the following sections, we will focus on VQA and examine its recent research progress.

2.4.2 Popular Visual Question Answering Datasets

In this section, for better understanding of tasks being addressed in VQA, we compare popular VQA/KB-VQA datasets with examples.

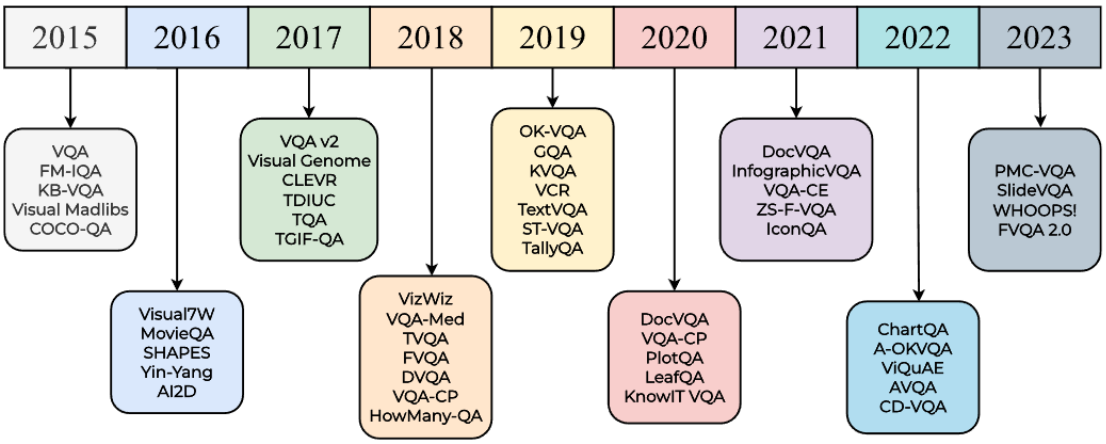


Fig. 2.35 Timeline of popular VQA datasets up to the end of 2023. The figure is from Ishmam et al. [122].

VQA datasets typically consist of data entries with the form: {image, question, answers}. A question is associated with the visual content of an image, with the correct answers provided by multiple human annotators. A more complete timeline of popular VQA datasets is presented in Fig. 2.35. Here, we introduce 7 popular VQA datasets, featuring different aspects of question answering:

VQA 2.0 [95]: a dataset containing open-ended questions about images. These questions require image understanding, language understanding, and commonsense knowledge to answer, e.g., "What color is the hydrant?"(answer: red) (the associated image is shown in Fig. 2.36), "What is being flown?"(answer: kite), and "What is the kid doing?"(answer: skateboarding).

Fig. 2.36 An example of the VQA 2.0 dataset. The associated question is "What color is the hydrant?".

GQA [120]:

a dataset featuring scene-graph-based questions over real-world images. An example of scene graphs is shown in Fig. 2.37. A potential question associated with this scene graph is "what color is the racket the girl is holding, yellow or green?". In this dataset, each image is associated with a scene graph of the image's objects, attributes and relations.

FVQA [311]: a VQA dataset which requires, and supports, much deeper reasoning. FVQA consists of questions that require external information to answer. For each answer to a question, a piece of supporting fact is provided, as shown in Fig. 2.38.

OK-VQA [214]: a benchmark where the image content is not sufficient to answer the questions, encouraging use of external knowledge resources. An example of the questions is shown in Fig. 2.39. Such questions require both image understanding and real-world knowledge to answer.

Infoseek [44] and E-VQA [218]: recently introduced KB-VQA benchmarks. Compared to traditional KB-VQA datasets such as F-VQA and OK-VQA, E-VQA and Infoseek present more challenging questions that require domain-specific knowledge for accurate responses. In contrast, F-VQA predominantly features questions that can be answered with simple, commonsense knowledge, and a significant proportion of OK-VQA questions do not necessitate external knowledge for correct answers, as highlighted by Schwenk et al. [272], Chen et al. [44], and Mensink et al. [218]. Illustrative examples are provided in Fig.2.40 and Fig.2.41.

2.4.3 Recent VQA Systems

In this section, we provide an overview of modern neural VQA systems. These systems can be categorised into two types based on their methodology: (1) bottom-up and top-down systems and (2) end-to-end systems.

Bottom-Up and Top-Down Systems

In a very influential work Anderson et al. [8] introduced the "bottom-up and top-down" architecture. This type of system typically consists of two components: one component is "bottom-up" and the other is "top-down".

The "bottom-up" module extracts task-agnostic, low-level visual features using feed-forward neural models. This module is trained independently on various vision tasks, such as object detection and image classification. By training this module on non-VQA datasets, it is possible to leverage richer data sources in object detection and image classification, such as Microsoft COCO [181] and Visual Genome [153]. Unlike VQA datasets, which are typically small in size, these image-only datasets can be produced on a larger scale due to the reduced

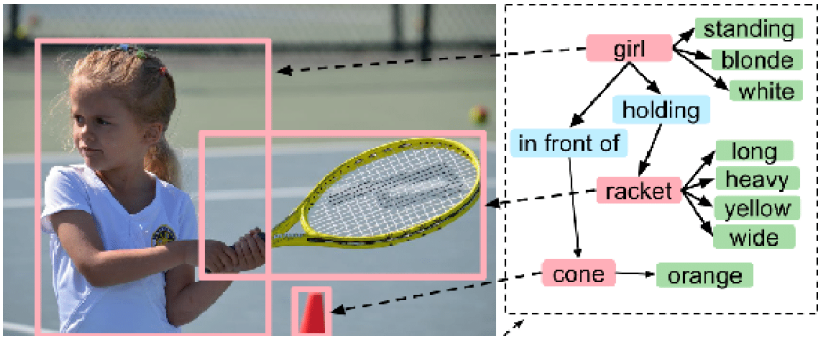


Fig. 2.37 An example of a scene graph. Red, green, and blue rectangles are for objects, attributes, and their semantic relations, respectively. The figure is from [1].

Fig. 2.38 An example of the FVQA dataset. The answer to the question is associated with a supporting fact.

Question : What country is named here?

Answer: tahiti
Answer Occurence: 5 / 5
Category: Vehicles and Transportation

Fig. 2.39 An example question from the OK-VQA dataset. The question requires both image understanding and real-world knowledge.

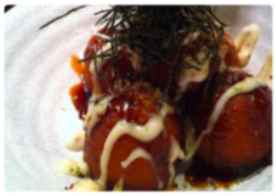
- **Q**: Who is the founder of this sport?
- **A**: Kanō Jigorō

- **Q**: What is the length of this bird in centimetre?
- **A**: 120-170

- **Q**: Who is the inventor of this object?
- **A**: James Gregory

- **Q**: What is the country of origin of this animal?
- **A**: Rhodesia

- **Q**: In which year did this building officially open?
- **A**: 1930



- **Q**: Which year was this food invented?
- **A**: 1935

- **Q**: What is the highest note this item can play?
- **A**: C8

- **Q**: Where is this plant native to?
- **A**: Ecuador

Fig. 2.40 Example questions from the Infoseek dataset. The questions require domain-specific knowledge. Q: Question; A: Answer.

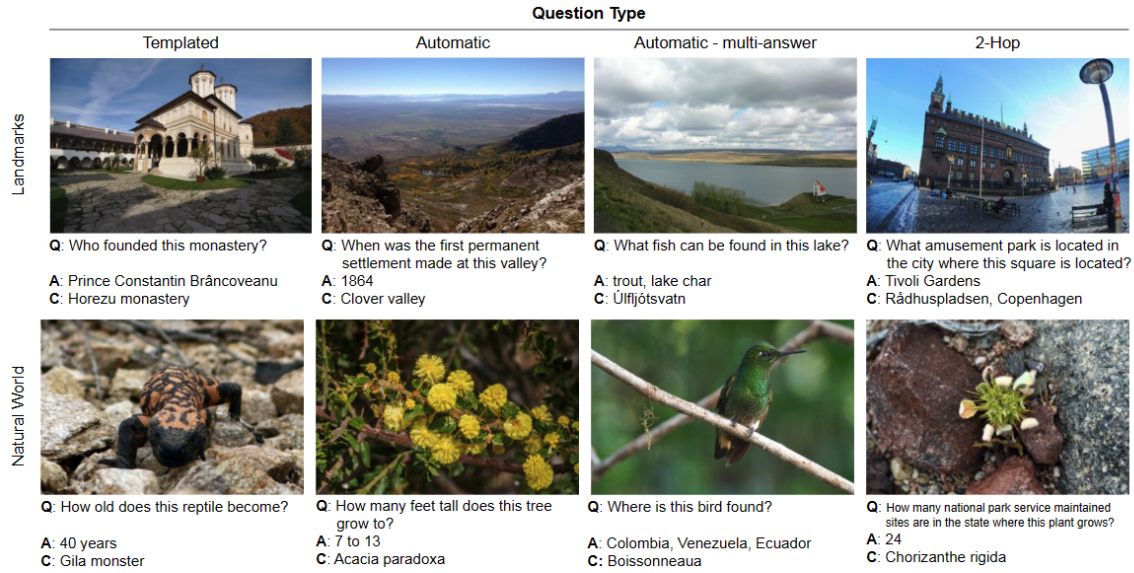


Fig. 2.41 Example questions from the E-VQA dataset. The questions require domain-specific knowledge. Q: Question; A: Answer. C: The caption of the associated ground-truth document.

complexity of annotation. Consequently, the "bottom-up" module is trained with large-scale task-agnostic data, enabling it to generate meaningful feature representations for images.

Recent research has employed pre-trained object detectors as the "bottom-up" module [8, 370], given that object detectors can extract objects, their attributes, and the semantic relationships between objects, which are more useful for question answering than a single representation of the entire image. For instance, both Bottom-up and Top-down [8] and VinVL [370] utilises Faster-RCNN [253] as the "bottom-up" module. VinVL pre-trained the module with four popular large public datasets, including Microsoft COCO [181] and Visual Genome [153], achieving strong performance on object detection.

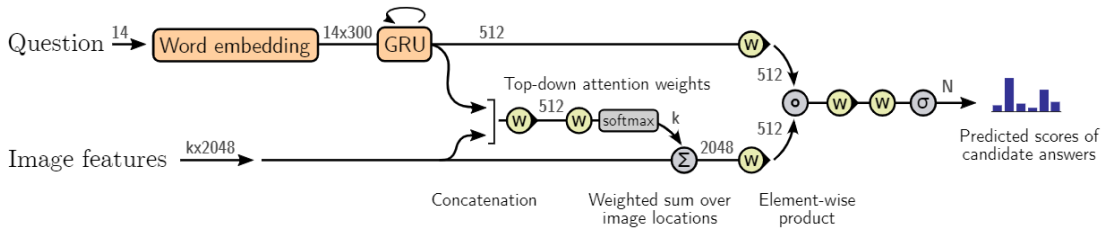


Fig. 2.42 The "top-down" module used in Anderson et al. [8].

The "top-down" module leverages the low-level features extracted by "bottom-up" module and performs answer generation. It takes in the task-agnostic image features (typically a list of regional features for each object detected in the image) and the question to predict

an answer. For example, as shown in Fig. 2.42, Bottom-up and Top-down [8] uses a Gated Recurrent Unit (GRU) to encode the question sentence into a question embedding, and the object features are aggregated selectively with an attention mechanism that attends to the question embedding. Then, the question embedding serves as a query to predict the score of each answer candidate. In Fig. 2.24, VinVL [370] uses pre-trained multi-layer Transformers called Oscar [175] to predict the answer. The input image features are extracted using a pre-trained object detector, which is also considered as a "bottom-up" module.

In addition to extracting visual features via object detectors or visual encoders, VQA systems can also utilise Scene Graphs for question answering [162, 231, 267, 344]. These Scene Graphs can be generated from the input image through Scene Graph Generation (SGG). SGG entails creating a structured representation of an image by identifying objects, their attributes, and the relationships between them [332, 297, 296, 91]. This graph-like structure aids in understanding the context and interactions within the image, facilitating tasks such as VQA, object detection/segmentation, and image captioning. SGG transforms visual data into a more interpretable format, capturing both the spatial and semantic connections among various elements in the scene, which can provide rich information for the question-answering component of VQA systems.

End-to-End Systems

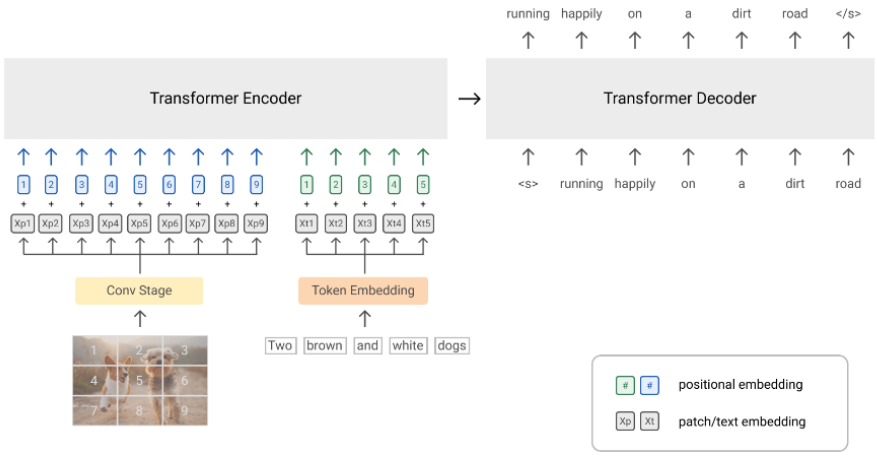


Fig. 2.43 The architecture of SimVLM [318]. This VQA system was trained end-to-end. The model was first pre-trained on large-scale web datasets for image-text inputs, as depicted in the figure. The input could be an image and its text description. Finally, the model is fine-tuned on downstream tasks such as VQA.

The "top-down and bottom-up" architecture has achieved great performance in VQA, but the system is split into two independent parts and thus joint optimisation is difficult. The

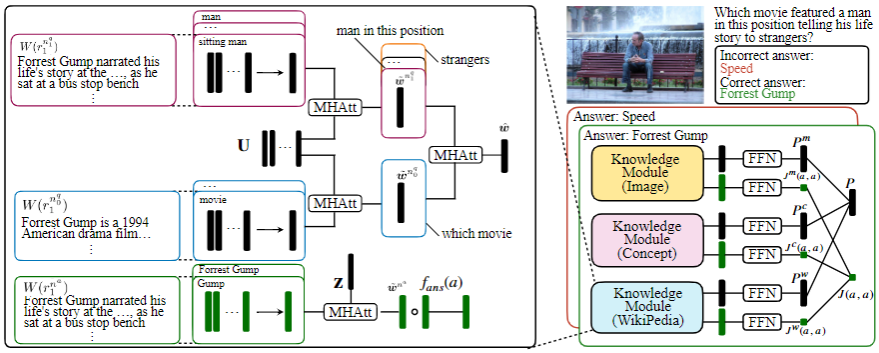


Fig. 2.44 The answer validation module of MAVEx [326]. It operates in three stages: (1) Answer Candidate Generation: Potential answers are proposed based on the given question and image. (2) Knowledge Retrieval: Relevant textual and visual information is gathered from sources like Wikipedia and Google Images, tailored to each candidate answer. (3) Answer Validation (which is shown in this figure): The candidates are evaluated against the retrieved multi-modal knowledge to validate the most accurate answer. The validation module is complicated and feature-engineered.

end-to-end approach trains a system as a whole: the visual feature extraction and downstream VQA tasks are trained with a global training objective. Some recent work has attempted to simplify the two-step independent training of modules into one. For example, SimVLM [318] was trained end-to-end, where the image and the associated question are directly given to a Transformer-based encoder-decoder model, as shown in Fig. 2.43. Instead of detecting objects and regions of interest in the image, SimVLM trained positional embeddings for each grid region of the image through large-scale pre-training, which is similar to the approach of Vision in Transformers (ViT) [72] on the image classification task.

LMM Models for VQA

More recently, LMMs have emerged as powerful multi-modal systems that are capable of solving a wide range of multi-modal tasks, including VQA. As introduced in Sec. 2.1.4, an LMM often consists of a visual encoder, a mapping network, and a language model. The visual encoder extracts task-agnostic features, and the mapping network connects the visual features with the language model. This architecture can also be attributed to "bottom-up and top-down" since the visual encoders are often pre-trained on vision-centric tasks and are frozen during LMM training.

The large-scale pre-training empowers the model's ability to answer visually-grounded questions, leading to superior performance on VQA tasks. Some examples of these systems are: LLaVA, QWen-VL, InternVL, and MiniGPT-v2 (see Sec. 2.1.4 for more introduction). In

contrast to traditional VQA systems, which are optimised solely for VQA tasks, these models utilise vast and diverse datasets and tasks during pre-training. This approach enables them to develop a rich and comprehensive understanding of both vision and language. Moreover, the pre-training process imparts a high degree of generalisability to these models, allowing them to adapt to a wide range of VQA scenarios without the need for extensive task-specific fine-tuning. As discussed in Sec. 2.1.4, various prompting methods can enhance multi-modal performance due to the strong zero-shot, few-shot, and reasoning capabilities of LLMs.

2.4.4 Knowledge-Aware VQA Systems

Knowledge-based VQA (KB-VQA) is a challenging VQA task. A recent trend of VQA is to build knowledge-aware systems that leverage external knowledge for answering questions. This is particularly important when questions require domain-specific, expert-level knowledge to answer. Datasets such as FVQA [311], OK-VQA [214], Infoseek [44], and E-VQA [218] provide questions that need external knowledge to answer.

Many systems have been developed to incorporate knowledge in the process of answer generation. For example, FVQA [311][3] performs database query explicitly using the elements extracted from both the image and the question. It retrieves sentences with supporting-facts to answer the questions in the FVQA dataset that they created. In particular, the OK-VQA dataset encourages use of outside knowledge for improving the system. Recent work on this dataset mainly focus on two types of data: (1) structured data in knowledge graphs (KG) such as ConceptNet [90]; (2) unstructured data such as passages from Wikipedia [326, 96, 36] and Google Search [208, 188, 183]. For example, VRR [208] used a passage retriever that retrieves relevant passages from a large text corpus that can help answering questions. The corpus was formed by Google Search results. In addition to extracting embeddings for questions and visual elements in the images with Transformers, KRISP [215] designed a "symbolic knowledge module" for matching ConceptNet KG entities with language/visual elements in the question. There are also very complicated systems such as MAVEx [326] and BreakDownVQA [324] where multiple knowledge sources are feature-engineered and fused for predicting an answer (Fig. 2.44).

Concurrent to the research presented in this thesis, some models demonstrated that simple text Transformers alone can achieve good performance in Knowledge-based VQA with the help of specialised vision models [328, 341, 306, 88, 183, 301]. For example, PICa [341] leverages GPT-3 for few-shot prompting of VQA, inspired by Tsimpoukelli et al. [306]. An offline Image-to-Text model is employed to generate image descriptors and the descriptions

[3] Here FVQA refers to the system they proposed along with the FVQA dataset

are appended to the question: "Please answer the following question: is the TV working or broken? a TV is on the grass." (the underlined content is generated by the offline image-to-text model). RA-VQA [183] (our work, introduced in Chapter 3) and TRiG [88] also leverage visual models to generate textual descriptions for images and then produce predictions with a Text Transformer. Plug-and-Play [301] determines several regions of interest with respect to the input question using Grad-CAM [274] and generate separate captions for the input image. An answer is then generated by a QA model using all captions. PromptCap [114] trains a specialised captioning model to generate more informative captions for question answering.

In terms of retrieval, this thesis leads a line of work that aims to improve KB-VQA performance through improving the retrieval performance on which the end-to-end performance depends. Lin and Byrne [183] (Chapter 3) enhances retrieval models through joint optimisation with the answer generator; Lin et al. [188] (Chapter 5) introduces multi-modal late-interaction retrieval models to achieve performance comparable to models of more than 7B parameters; Lin et al. [190] (Chapter 6) scales up the multi-modal retrieval models to achieve state-of-the-art retrieval performance on popular KB-VQA datasets.

LLMs have been used extensively since the release of GPT-3. Many systems incorporate LLMs/LMMs as answer generators [328, 341, 306, 88, 183, 301], while there is another line of work treat LLMs as implicit knowledge bases. For example, Gui et al. [96, KAT], Lin et al. [180, REVIVE] and Shao et al. [277, Prophet] use LLM to generate answer candidates which are then jointly considered by another question answering module.

In recent years, LMMs with billions of parameters, pre-trained on extensive datasets, have demonstrated remarkable performance on KB-VQA tasks, even without relying on external knowledge bases [115, 73, 38, 40]. For instance, PaLM-E [73], an extensive ensemble model with 562 billion parameters, has attained state-of-the-art performance on the OK-VQA benchmark.

2.5 Table Question Answering

In this section, we provide a concise overview of the recent advancements in Table Question Answering (TableQA).

TableQA involves the task of automatically answering questions using semi-structured tables. This task necessitates extracting relevant table content to respond to a user's query and generating either a list of cell values or numerical values derived from selected table regions via aggregation functions (e.g., SUM, which adds up the values in the selected cells), commonly referred to as denotation [237]. An illustrative example is presented in Fig. 2.45. This example presumes the availability of the table; however, the problem becomes

Table

Rank	Name	No. of reigns	Combined days
1	Lou Thesz	3	3,749
2	Ric Flair	8	3,103
3	Harley Race	7	1,799
4	Dory Funk Jr.	1	1,563
5	Dan Severn	2	1,559
6	Gene Kiniski	1	1,131

Example questions

#	Question	Answer	Example Type
1	*Which wrestler had the most number of reigns?*	Ric Flair	Cell selection
2	*Average time as champion for top 2 wrestlers?*	AVG(3749,3103)=3426	Scalar answer
3	*How many world champions are there with only one reign?*	COUNT(Dory Funk Jr., Gene Kiniski)=2	Ambiguous answer
4	*What is the number of reigns for* Harley Race?	7	Ambiguous answer
5	*Which of the following wrestlers were ranked in the bottom 3?*	{Dory Funk Jr., Dan Severn, Gene Kiniski}	Cell selection
	Out of these, who had more than one reign?	Dan Severn	Cell selection

Fig. 2.45 Given a table (left), several questions can be asked that might be answered by content of some cell, or aggregation of multiple cells (right). The figure is from Herzig et al. [106].

significantly more challenging if a valid table must first be identified from a large corpus using a retrieval method.

We are primarily interested in two main task definitions for TableQA: (1) closed-domain TableQA and (2) open-domain TableQA. In closed-domain TableQA, an example in the training set is represented as a triple (q, T, a), where q denotes an utterance, T denotes the table used to extract information to answer the utterance, and a represents the corresponding set of denotations answering the question q. The objective is to train a model that maps a new utterance q^* to a model z such that, when applied to table T^*, it produces the correct denotations a^*. These denotations can either be the value of one or more table cells, known as cell selection, or an aggregated value from a subset of table cells, such as SUM or MAX (finding the max value in the selected cells), which results in a scalar answer.

In contrast, open-domain TableQA involves training a model on a set of triples (q, T, a) and a corpus of tables C. The goal is to develop a model that, given a new utterance q^* and the corpus C, returns the correct answer a by finding the tables that contain the needed information.

2.5.1 Popular Datasets

The most popular closed-domain TableQA datasets include WikiTQ (WikiTableQuestions) [237], WikiSQL [381], and SQA [123]. WikiTQ consists of complicated questions regarding tables. The answers and questions were annotated by crowd workers. The questions cover complex table question answering, requiring the ability of comparisons, superlatives, value aggregation, arthmetic operations to answer. WikiSQL contains questions, SQL (Structured Query Language) operations, and their resulting outcome when performed on a database. The corresponding questions were annotated by crowd workers. SQA was constructed by asking

crowd workers to decompose some highly compositional WikiTQ questions into multiple smaller, decomposed questions. These three datasets have been widely used to assess recent TableQA systems.

In open-domain TableQA, the datasets can often be created from closed-domain TableQA datasets by hiding the association between questions and their corresponding tables. For instance, NQ-TABLES [107] extracted table-related questions from NaturalQuestions [155] and subsequently decoupled the questions from the related tables. E2E-WTQ [235] comprises look-up questions that require cell selection operations and is a subset of WikiTableQuestions. The train/validation/test splits in E2E-WTQ are identical to those in WikiTableQuestions, with questions restricted to those that do not require aggregation across multiple table cells.

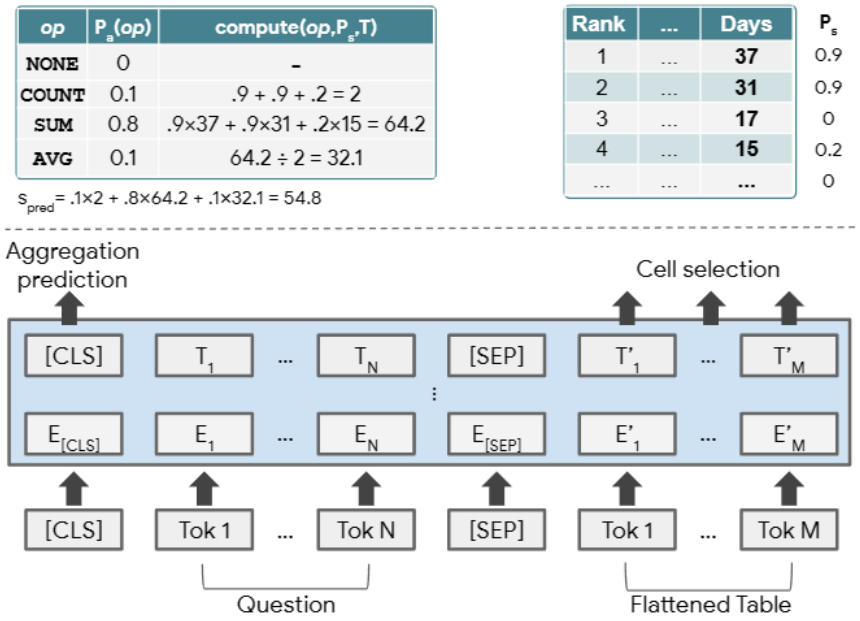


Fig. 2.46 The TaPas model. The model uses a Transformer encoder to simultaneously predict the aggregation function using the token representations of '[CLS]' and predict the relevant cells with the remaining token representations. Then the selected aggregation function (such as SUM) is applied to the selected table cell values.

2.5.2 Understanding Structured Tables with Transformers

To harness the capabilities of powerful Transformers, particularly those pre-training techniques and models within the domain of natural language processing, researchers have investigated multiple approaches to transform structured tables into text-like sequences. These methods aim to ensure that text Transformers can process the data effectively while maintaining an awareness of the inherent table structure.

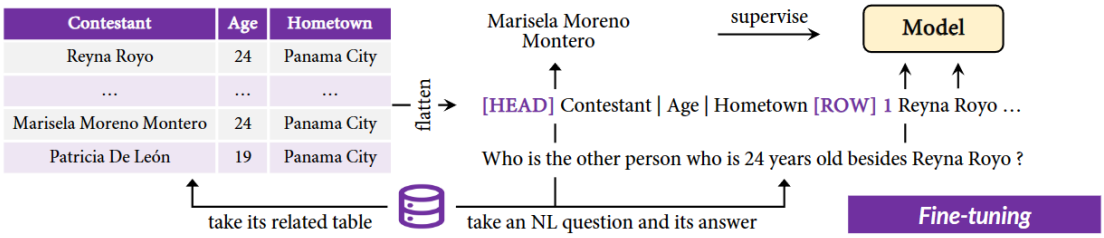


Fig. 2.47 The fine-tuning process of the TaPEx model. It was first pre-trained on large-scale synthetic SQL data and then fine-tuned on downstream TableQA tasks (shown in the figure). The model leverages an encoder-decoder Transformer to directly generate the answer. The table is linearised/flattened into text sequences with rows and columns separated by special tokens and colons.

For instance, TaPas (as shown in Fig. 2.46) utilises only the text within cells, but tracks the position of each token in the table and learns position, cell, and row embeddings to combine with the token embeddings. Specifically, they add trainable embedding layers to incorporate position embeddings, segment embeddings, column embeddings, row embeddings, and rank embeddings, which provide structural information such as the row and column indices of all table cells.

Building on top of TaPas, TableFormer [339] achieves more robust performance against column and row order perturbations by implementing several modifications, including (1) removing row and column embeddings that may introduce spurious order biases; (2) using per-cell positional embeddings; and (3) adding additional structure-aware attention biases to self-attention operations to make the model aware of cells in the same rows and columns.

TaBERT [353] selects the top-K rows with the highest n-gram overlap with the question and then performs per-row encoding on the selected rows. These row representations are subsequently passed to a vertical self-attention layer to extract column representations. The vertical self-attention mechanism ensures that the resulting column representations are more informative and relevant to the input question, effectively capturing the context and relationships within the table.

An alternative research direction favours more straightforward methodologies. TaPEx, as shown in Fig. 2.47, linearises tables into text sequences using a straightforward yet effective strategy: rows are separated by the special token '[ROW] row_index' and cells within a row are separated by 'l'. They demonstrate that TaPEx possesses the capability to comprehend table structures, even in the absence of specialised designs tailored for these structures [200].

In table retrieval, akin to approaches in closed-domain TableQA, recent work has designed models to parse complex tabular structures [107, 309], or by simply linearising tables with interleaving tokens to preserve their structure [236, 316].

2.5.3 Popular TableQA Models

In closed-domain TableQA, recent methodologies transform structural tables into text sequences, enabling the direct processing of structured data by Text Transformers. Earlier models generated commands in logical forms, such as SQL queries, executable over tables [357, 179, 333, 359]. In contrast, recent state-of-the-art models predict answers directly from the input question and table through either classification [106, 339] or autoregressive generation [200, 135]. The most representative models for classification and autoregressive generation are TaPas [106] and TaPEx [200], respectively. TaPas (Fig.2.46) employs a Transformer encoder to simultaneously predict the aggregation function using the token representations (or hidden states at the model output) of '[CLS]' (the first token of the Transformer encoder, as depicted in Fig. 2.46) and identify the relevant cells with the representations of the remaining tokens. To determine the relevant cells, the token representations within each cell are averaged and subsequently processed through a linear layer to obtain each cell's relevance to the query. Following this, an empirical threshold value is applied to select a subset of cells deemed relevant. The selected aggregation function (e.g., SUM) is then applied to the chosen table cell values. TaPEx (Fig.2.47), representing another line of research, predicts answers directly using an encoder-decoder Transformer.

Existing research on open-domain TableQA is relatively sparse compared to closed-domain TableQA. The most effective systems are based on a retriever-reader pipeline [107, 236]. For instance, Herzig et al. [107] utilise TaPas to initialise both a retriever and the reader similar to the bi-encoder retriever models in DPR (Sec. 2.2.1). T-RAG [236] employs DPR to retrieve rows/columns by decomposing the table and generates the answer through a sequence-to-sequence reader [166, BART], applying the RAG [167][4] loss to refine the retriever with implicit signals during end-to-end fine-tuning. Unlike DTR and T-RAG, CLTR [235] identifies the answer cell by intersecting the top-scored rows and columns. The score for each row/column is computed by a Transformer encoder, which takes the concatenation of the question and the row/column as input and outputs a score representing the probability of containing the answer. This approach has an obvious limitation in that it can not handle questions that require multiple cells to answer. In Chapter 8, we provide an in-depth analysis of the performance of these systems, presenting a novel system and conducting a comparative evaluation with these existing ones.

[4]Here RAG is the name of the proposed approach, introduced in Sec. 2.3.1 and Fig. 2.29.

2.6 Summary

In this chapter, we delved into recent advancements closely linked to Retrieval-Augmented Multi-Modal Systems.

In Sec.2.1, we outline the recent progress made in the development of Large Language Models and Large Multi-modal Models, which serve as the bedrock for the research presented in this thesis.

Following this, we pivot to present recent breakthroughs in Information Retrieval (Sec.2.2) and Retrieval Augmented Generation (Sec. 2.3). This encompasses advancements ranging from the text-only setting to the multi-modal setting, which have inspired the development of the retrieval-augmented multi-modal systems in this thesis.

Subsequently, we introduced the two core tasks addressed herein: Visual Question Answering (Sec.2.4) and Table Question Answering (Sec.2.5).

In the next chapter, we will introduce the Retrieval Augmented Visual Question Answering (RA-VQA) framework, which utilises techniques from Large Language Models, Information Retrieval, and Retrieval Augmented Generation.

Chapter 3

Retrieval Augmented Visual Question Answering Framework

3.1 Introduction

As introduced in the last chapter, Visual Question Answering (VQA) is a challenging problem that lies at the intersection of Computer Vision, Natural Language Processing, and Information Retrieval. VQA is particularly challenging when the answer to the question is not directly available in the image. In *Knowledge-based* VQA (KB-VQA), the VQA system must access external knowledge sources to find a correct and complete answer. The Ouside-Knowledge VQA task (OK-VQA) [214] consists of questions that requires general knowledge and simple inference to answer (Fig. 3.1). Such questions are even hard for humans. Unlike other KB-VQA datasets (e.g. FVQA [311], introduced in Sec. 2.4.2) which provide an associated knowledge base, OK-VQA encourages using any outside knowledge in answering questions. This chapter focuses on this challenging task, and we propose a novel framework, Retrieval Augmented Visual Question Answering (RA-VQA), that achieves excellent results on the OK-VQA dataset. We further verify its generalisability on FVQA.

The RA-VQA framework forms the foundation for the work presented in subsequent chapters. Specifically, the studies discussed in Chapters 4 through 6 and in Chapter 8 are all based on this framework.

The need to adapt and refresh knowledge sources motivates the study of KB-VQA systems that can extract knowledge from both structured (e.g. ConceptNet [288]) and unstructured knowledge representations (e.g. Wikipedia passages). Recent designs [208, 88] approach VQA in two distinct steps: (1) *Knowledge Retrieval* extracts relevant documents from a large knowledge base; (2) *Answer Generation* produces an answer from these documents.

Question : Which sesame street character would eat this?
Answer: cookie monster

Fig. 3.1 OK-VQA contains questions whose answer cannot be found within the image.

Knowledge Retrieval can be done via Dense Passage Retrieval (DPR) [143], which consists of a question encoder and a document encoder (both Transformer-based) that encode questions and documents into separate dense representations. The DPR system is trained to assign higher query-specific scores to documents intended to be helpful in answering questions, so that document sets can be retrieved and passed to Answer Generation for each given query.

Knowledge Retrieval based on DPR is powerful but has some readily observed limitations, particularly in model training. Firstly, whether a retrieved document is useful in answering a question cannot be easily determined, even if an answer is provided. Prior work [242, 208] has addressed this problem using "*Pseudo Relevance Labels*". Pseudo Relevance Labels mark a document as relevant if it contains any target answers of the question. However, these are only a weak signal of potential document relevance and may encourage DPR to retrieve misleading documents. Secondly, the document retriever and answer generator are trained separately. To ensure that the answer generator sees relevant documents in training, systems can retrieve large numbers of documents (∼50+) [88, 96], but at the cost of slower training and more GPU (Graphics Processing Unit) usage, and also possibly presenting misleading material to the answer generator.

Joint training of the retriever and answer generator offers a solution to these problems. The aim is twofold: (1) to improve the retrieval of documents truly relevant to providing a given answer; and (2) to reject documents with pseudo relevance but not true actual relevance.

RAG [167] (Sec. 2.3.1[1], Fig. 2.29) has shown that end-to-end joint training of a DPR-based QA system can outperform baseline two-step systems. A notable feature of RAG is a loss function that incorporates marginalised likelihoods over retrieved documents such that the training score of a document is increased whenever it improves prediction.

However, in preliminary OK-VQA experiments we found that RAG did not perform well. Our investigations found that a good portion of OK-VQA training questions are answerable in 'closed-book' form (i.e. using pre-trained models such as T5 [248]) with information

[1]To avoid confusion with Retrieval Augmented Generation, in Chapter 3 to Chapter 8, the term Retrieval Augmented Generation is consistently used without abbreviation.

extracted only from the image, with the unintended consequence that the RAG loss function awards credit to documents that did not actually contribute to answering a question. We also found that difficult questions that are unanswerable with the knowledge available to retrieval were more prevalent in OK-VQA than in the Open-domain QA datasets (e.g. Natural Questions [156]) on which RAG was developed. In both of these scenarios, the RAG loss function leads to counter-intuitive adjustments to the document scores used in training the retrieval model, leading to decreased VQA performance.

Motivated by these findings, we propose a novel neural-retrieval-in-the-loop framework for joint training of the retriever and the answer generator. We formulate a loss function that avoids sending misleading signals to the retrieval model in the presence of irrelevant documents. This formalism combines both pseudo relevance labels and model predictions to refine document scores in training. We find significantly better performance on OK-VQA compared to RAG. In this chapter:

- We present a novel joint training framework **R**etrieval **A**ugmented **V**isual **Q**uestion **A**nswering (RA-VQA) for Knowledge Retrieval and Answer Generation that improves over RAG and two-step baseline systems based on DPR [143].
- We investigate visual features transformed into 'language space' and assess their contribution to OK-VQA performance.
- We study the role of document retrieval in KB-VQA and evaluate its interaction with retrieval-augmented generation. We also show that retrieval becomes more efficient in joint training, requiring retrieval of relatively few (~ 5) documents in training.
- We implement the framework on the FVQA dataset and report excellent performance, which demonstrates the generalisability of our proposed framework to other tasks.

The code, data, and pre-trained model weights have been released at: https://github.com/LinWeizheDragon/Retrieval-Augmented-Visual-Question-Answering/tree/legacy_v1.

3.2 Related Work

This section provides an overview of the related work up to April, 2022, the time of this study.

3.2.1 Open-domain QA systems

Open-domain QA systems are designed to answer questions from conventional QA datasets such as Natural Questions [156]. The knowledge needed to answer questions can be in

pre-trained models [254], knowledge-graphs (KGs) [178, 86, 209, 259] or document collections [30, 124, 99, 161, 167]. In retrieval-based systems, as detailed in Sec. 2.3.1, the retriever and the question-answering component can be jointly optimised. This joint training approach can be integrated with extractive question answering, as demonstrated by REALM [99] and ORQA [161], as well as with generative answer generation, as exemplified by RAG [167] (Fig. 2.29). RAG is similar to our work in that it also employs generative answer generation and trains both the answer generator and the retriever simultaneously, albeit with a distinct loss function. We replicate RAG as a baseline for comparison with our work.

3.2.2 VQA Systems

Modelling vision and language is central to VQA. Models can aggregate visual and textual features via cross-modality fusion [362, 286, 363, 133, 97]. Systems can also be pre-trained on large vision-and-language collections [130] and then fine-tuned for VQA tasks [294, 45, 87, 175, 318, 370, 173] with VQA datasets such as VQA 2.0 [11] (introduced in Sec. 2.4.2).

As discussed in Sec. 2.4.4, KB-VQA systems require access to data beyond vision and language. They can access both structured data, such as ConceptNet and other Knowledge Graphs (KGs) [227, 90, 168, 326, 215], as well as unstructured data such as Wikipedia passages [326, 88, 96]. Up to the time of this study, a variety of multi-modal approaches have been explored to access external knowledge, such as ConceptBERT [90], KRISP [215], MAVEx [326], and VRR [208] (discussed in Sec. 2.4.2). Our work adopts a similar retriever-reader pipeline with a focus on the joint training of the retriever and reader/generator. The proposed framework is capable of managing both unstructured passage collections and knowledge-graph data that is transformed into flattened texts (Sec. 3.4.5).

Prior work has established that images can be transformed into text so that large pre-trained language-based Transformers (e.g. BERT, GPT-2, and T5, as discussed in Sec. 2.1.3) can be applied to VQA tasks [208, 341]. Systems can be based on straightforward image caption that is generated by image captioning models. In contrast, in our work, we have found improvements by introducing additional visually-grounded textual features, such as object tags detected in the images (Sec. 3.4.5).

We also note unpublished contemporaneous work on OK-VQA, such as TRiG [88], PICa [341], and KAT [96], and we have discussed them in Sec. 2.4.4.

3.3 Method

In this section, We present our RA-VQA framework that consists of: (1) Vision-to-Language Transformation (Sec. 3.3.1); (2) Weakly-supervised Dense Passage Retrieval (Sec. 3.3.2); (3) Joint Training of Retrieval and Answer Generation (Sec. 3.3.3).

3.3.1 Vision-to-Language Transformation

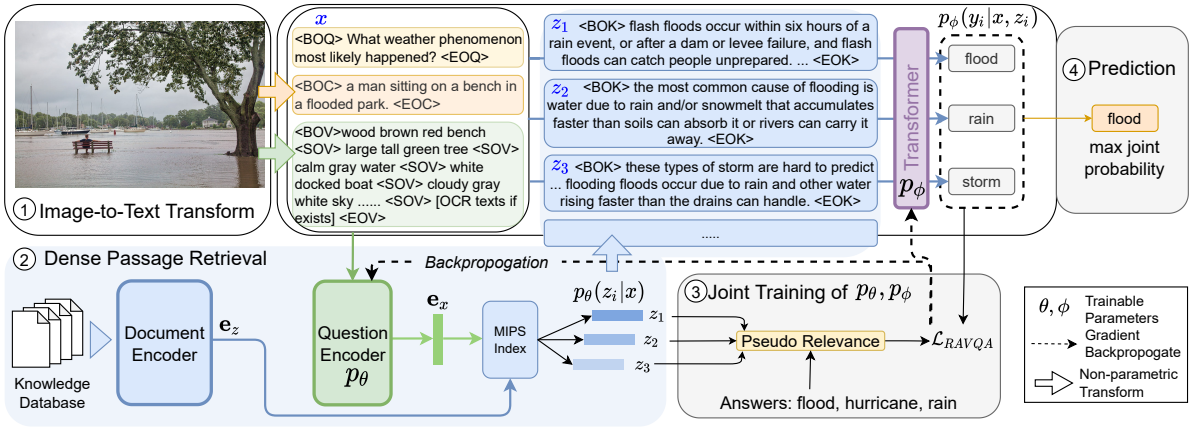


Fig. 3.2 Model overview. (1) Using object detection/image captioning/Optical Character Recognition to transform visual signals into language space. (2) Dense Passage Retrieval retrieves documents that are expected to be helpful from the knowledge database; (3) Training the retriever p_θ and the answer generator p_ϕ together using our proposed RA-VQA loss. (4) The answer with highest joint probability $p_\theta(z_i|x)p_\phi(y_i|x,z_i)$ is selected.

In RA-VQA, each image is represented by visual objects and their attributes, image captions, and any text strings detected within the image. We use an object detection model VinVL [370] that was pre-trained on large object detection datasets to extract visual elements and their attributes (e.g. color and material).

Formally, for an image I we use VinVL to extract a set of visual objects $\{o_i\}$, along with a set of text attributes for each visual object $\{a_{i,j}\}$. Visual objects and their attributes are extracted by VinVL at confidence thresholds 0.8 and 0.6, respectively.

Image captioning is performed to extract relationships and interactions among visual elements such as "a woman holding a knife cuts a cake". The pre-trained captioning model Oscar+ [370] is applied to process visual features extracted from the VinVL model to generate a caption for the image. To answer questions related to text strings in images (e.g. "which

language is the book written in?"), Google OCR (Optical Character Recognition) APIs[2] are used to extract text strings from each image.

Hence, a VQA training set $\{(I, q, \mathscr{S})\}$, where $\mathscr{S}$ is a set of answers to a question q about I, can be transformed into a text-only training set $\mathscr{T} = \{(x, \mathscr{S})\}$ that we use for RA-VQA. The string x contains all the text features extracted from the image (the question, the textual attributes for each identified visual object, the generated caption, and any OCR'd text), with special tokens marking the start and end of each type of feature (Fig. 3.2).

3.3.2 Weakly-supervised Dense Passage Retrieval

Dense Passage Retrieval in RA-VQA consists of a query encoder $\mathscr{F}_q$ and a document encoder $\mathscr{F}_d$, both as Transformer-like encoders. The aim is to retrieve K documents from an external knowledge database $\mathscr{Z} = \{z_i\}_{i=1}^{N^d}$ (e.g. Wikipedia passages) that are expected to be useful for answering a question. DPR encodes questions and documents separately into dense feature vectors $\mathscr{F}_q(x) \in \mathbf{R}^h$ and $\mathscr{F}_d(z) \in \mathbf{R}^h$. A scoring function is used to retrieve documents for each question as the inner product between the representations of x and z

$$r(x,z) = \mathscr{F}_q^\top(x)\mathscr{F}_d(z). \tag{3.1}$$

RA-VQA training aims to maximise $r(x,z)$ when document z is relevant to answering the question. As discussed in Sec. 3.1, the relevance between q and z cannot be easily obtained and "pseudo relevance labels" serve as a proxy. We use a pseudo relevance function $H(z, \mathscr{S})$ which is 1 if z contains an answer in $\mathscr{S}$ (by string match), and 0 otherwise.

For each question-answer pair $(x, \mathscr{S})$ one positive document $z^+(x)$ is extracted for training. In-batch negative sampling is used: all documents in a training batch other than $z^+(x)$ are considered to be negative for $(x, \mathscr{S})$ [143]. Denoting the negative documents as $\mathscr{N}(x, \mathscr{S})$ and the score of the positive document as $\widehat{r}^+(x)$ leads to the DPR loss $\mathscr{L}_{DPR}$:

$$-\sum_{(x,\mathscr{S})\in\mathscr{T}} \log \frac{\exp\left(\widehat{r}^+(x)\right)}{\exp\left(\widehat{r}^+(x)\right) + \sum_{z\in\mathscr{N}(x,\mathscr{S})} \exp\left(\widehat{r}(x,z)\right)}. \tag{3.2}$$

3.3.3 Joint Training of Document Retrieval and Answer Generation

Given a full query string x extracted from the image-question pair (I, q), DPR returns the K highest scoring documents $\{z_k\}_{k=1}^K$. The score assigned by the document retriever $p_\theta(\cdot|x)$ to

[2] https://cloud.google.com/vision/docs/ocr

a retrieved document is

$$p_\theta(z_k|x) = \frac{\exp(\widehat{r}(x,z_k))}{\sum_{j=1}^{K}\exp(\widehat{r}(x,z_j))}. \tag{3.3}$$

Open-ended answer generation for each retrieved document z_k is performed with a generative model, such as T5 (Sec. 2.1.3), with parameters ϕ:

$$y_k = \underset{y}{\text{argmax}}\, p_\phi(y|x,z_k). \tag{3.4}$$

For each document z_k retrieved for a training item $(x,\mathscr{S})$, we select the most popular human response s_k^* from $\mathscr{S}$ such that s_k^* is contained in z_k; in the case that z_k does not contain any answer, the most popular answer $s^* \in \mathscr{S}$ is selected $s_k^* = s^*$. We identify two subsets of the retrieved documents $\{z_k\}_{k=1}^K$ based on pseudo relevance labels and model predictions:

$$\begin{aligned}\mathscr{P}^+(x,\mathscr{S}) &= \{k : y_k = s_k^* \wedge H(z_k,\mathscr{S}) = 1\};\\ \mathscr{P}^-(x,\mathscr{S}) &= \{k : y_k \neq s_k^* \wedge H(z_k,\mathscr{S}) = 0\}.\end{aligned} \tag{3.5}$$

$\mathscr{P}^+$ are indices of pseudo relevant documents that also help the model generate popular answers whereas $\mathscr{P}^-$ identifies documents not expected to benefit answer generation.

Joint training of retrieval and answer generation is achieved by optimising a loss function $\mathscr{L}_{RA-VQA}$ that reflects both model predictions and pseudo relevance:

$$\begin{aligned}-\sum_{(x,\mathscr{S})\in\mathscr{T}} \Big(&\sum_{k=1}^{K}\log p_\phi(s_k^*|x,z_k)\\ &+\sum_{k\in\mathscr{P}^+(x,\mathscr{S})}\log p_\theta(z_k|x) - \sum_{k\in\mathscr{P}^-(x,\mathscr{S})}\log p_\theta(z_k|x)\Big).\end{aligned} \tag{3.6}$$

The first term in the loss improves answer generation from queries and retrieved documents, taken together. The remaining terms affect document retrieval: the second term encourages retrieval of documents that are not only pseudo relevant but also lead to production of correct answers, while the third term works to remove irrelevant items from the top ranked retrieved documents.

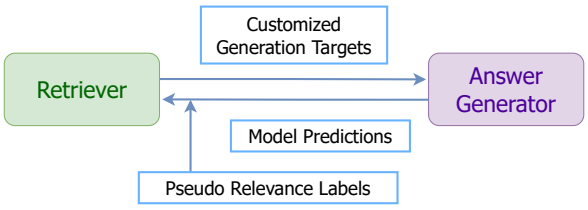


Fig. 3.3 Information flow between the retriever and the answer generator.

The information flow is demonstrated in Fig. 3.3. Retrieval and generation complement each other in training: pseudo relevance labels and model predictions provide positive and negative signals to improve retrieval, and the improved retrieval leads to improved answer generation by training towards s_k^*, a customised generation target for each retrieved document z_k.

3.3.4 RA-VQA Generation

Given an image query (I,q), a full query x is created (Sec. 3.3.1) and answer generation searches for the answer with the highest joint probability:

$$\begin{aligned} \{z_k\}_{k=1}^{K} &= \underset{z}{\text{argmax}^K}\, p_\theta(z|x), \\ \widehat{y},\widehat{z} &= \underset{y,z_k}{\text{argmax}}\, p_\phi(y|x,z_k)\; p_\theta(z_k|x). \end{aligned} \tag{3.7}$$

The joint probability comprises both the generation confidence $p_\phi(y|x,z_k)$ and the retrieval confidence $p_\theta(z_k|x)$. Consequently, the retrieval confidence significantly influences answer generation, in contrast to certain prior works, such as Luo et al. [208], where the retrieval confidence is not factored into the selection of candidate answers.

3.3.5 Pre-Computed FAISS Document Indices

Since repeated computation of embeddings for all documents is costly, we follow Lewis et al. [167] who find that it is enough to train only the question encoder $\mathscr{F}_q$ and leave document encoder $\mathscr{F}_d$ fixed. As shown in Fig. 3.2, document embeddings are pre-extracted with a pre-trained DPR document encoder. The FAISS system [138] is used to index all document embeddings which enables fast nearest neighbour search with sub-linear time complexity. In training, question embeddings are generated dynamically and documents with highest scores are retrieved using the pre-computed index.

3.4 Experiments

3.4.1 Datasets and RA-VQA Configurations

OK-VQA [214] was the largest knowledge-based VQA dataset at the time of this study. It consists of 14,031 images and 14,055 questions. These questions are split into a training set

(9,009 questions) and a test set (5046 questions). In addition to understanding images and questions, external knowledge sources are needed to answer questions.

As outside knowledge we use the knowledge corpus collected by Luo et al. [208] from Google Search. We use the corpus `GS-full` which consists of 168,306 documents covering training and test questions. In contrast, `GS-train` contains documents relevant to OK-VQA training set questions only. Unless otherwise specified, we use `GS-full` in experiments.

Pre-training: We start with pre-trained versions of BERT-base and T5-large (introduced in Sec. 2.1.3) as the document retriever and the answer generator, respectively. The retriever was refined by training it on `GS-full` under the DPR loss (Equation 3.2) with pseudo relevance labels released by Luo et al. [208]. The already strong retriever serves as a good starting point for all DPR-based models presented in this paper (including RA-VQA and our replication of baselines in the literature).

OK-VQA Fine-tuning: Our **RA-VQA** framework trains the answer generator and the retriever jointly under Equation 3.6.

We also report on variants of RA-VQA, to investigate the contribution of various model components to overall performance:

RA-VQA-NoDPR omits retrieval entirely so that answers are generated by the fine-tuned T5 alone. RA-VQA generation in Equation 8.6 simplifies to

$$\widehat{y}_{NoDPR} = \underset{y}{\operatorname{argmax}}\, p_\phi(y|x), \tag{3.8}$$

where p_ϕ is the answer generator (the T5 model, specifically).

RA-VQA-FrDPR leaves the retriever frozen after pre-training and fine-tunes the generator only.

RA-VQA-NoPR is a version of RA-VQA in which document retrieval is trained only with model predictions. The loss function is as Equation 3.6, but with positive and negative document sets defined as

$$\begin{aligned} \mathscr{P}^{+}_{NoPR}(x,\mathscr{S}) &= \{k : y_k = s^*_k\}; \\ \mathscr{P}^{-}_{NoPR}(x,\mathscr{S}) &= \{k : y_k \neq s^*_k\}. \end{aligned} \tag{3.9}$$

RA-VQA-NoCT replaces the customised generation targets by the single most popular response (s^*_k becomes s^* in Equation 3.6) so that the generator is trained to produce the same answer from every retrieved document.

3.4.2 Evaluation

The general principles for choosing metrics to assess retrieval-augmented systems are discussed in Section 1.3. The following metrics are applied to assess the quality of individual answers generated and documents retrieved.

Average scores are then computed over the evaluation set. The average of 3 runs with different seeds is reported.[3]

Answer Evaluation

VQA Score: We follow Marino et al. [214] to compute VQA Scores using pre-processed human annotations $\mathscr{S}$:

$$\text{VQAScore}(y, \mathscr{S}) = \min\big(\frac{\#_{\mathscr{S}}(y)}{3}, 1\big), \tag{3.10}$$

where $\#_{\mathscr{S}}(y)$ is the number of annotators who answered y. This score ensures that a model is partially rewarded even if it generates one of the less popular answers from amongst the human responses, but the benefits of popularity is capped at 3.

Exact Match (EM) treats annotated answers equally: $\text{EM}(y, \mathscr{S}) = \min(\#_{\mathscr{S}}(y), 1)$.

Retrieval Evaluation

Following Luo et al. [208], we use pseudo relevance to ascertain whether the retrieved documents are relevant to the response. It concerns pseudo relevance instead of actual relevance but is still a reasonable metric for retrieval evaluation.

Pseudo Relevance Recall (PRRecall)@K measures how likely the retrieved K documents contains at least one positive document:

$$\text{PRRecall@K} = \min\big(\sum_{k=1}^{K} H(z_k, \mathscr{S}), 1\big). \tag{3.11}$$

Integrated System Evaluation

The above methods evaluate retrieval and answer generation as separate processes. We propose additional metrics that assess how the two processes behave in an integrated VQA system. To the best of our knowledge, these metrics have not yet been adopted in another paper in this field.

[3] See Sec. 2.1.2 for the discussion of random seeds.

The **Hit Success Ratio (HSR)** counts questions that require external knowledge to answer:

$$HSR = \mathbb{1}\left\{\widehat{y} \in \mathscr{S} \wedge \widehat{y}_{NoDPR} \notin \mathscr{S}\right\}. \tag{3.12}$$

HSR reflects the value of incorporating external documents into answer generation.

By contrast, **Free Success Rate (FSR)** counts questions that can be answered without external knowledge.

$$FSR = \mathbb{1}\left\{\widehat{y} \in \mathscr{S} \wedge \widehat{y}_{NoDPR} \in \mathscr{S}\right\}. \tag{3.13}$$

A high FSR suggests a model can generate correct answers 'freely' without being distracted by retrieved documents if they are not needed.

We also assess performance as a function of the number of documents retrieved during training and testing, $\mathbf{K_{train}}$ and $\mathbf{K_{test}}$. In practice, K_{train} has the greater effect on GPU usage, since a large K_{train} requires at least K_{train} forward passes for each question and an Adam-like optimizer must compute and store the associated gradients [149]. In contrast, GPU memory required during testing is significantly less, as there is no optimizer involved. We are in particular interested in the ability of knowledge-augmented systems that can be robustly trained with small K_{train} while yielding improved performance with large K_{test}.

3.4.3 Training Details and Artifacts

The explanation for optmizers, schedulers, and relevant concepts can be found in Sec. 2.1.2.

We use Adam optimizer [149] to train our models. In pre-training the DPR component (Sec. 2.2.1), the retriever was trained for 6 epochs with a constant learning rate 10^{-5}. In training the entire framework, the learning rates are 10^{-5} for the retriever, and 6×10^{-5} for the answer generator, linearly decaying to 0 after 10 epochs. In the training of RA-VQA-NoDPR and TRiG* (introduced in Sec. 3.4.4), the initial learning rate is 6×10^{-5}. Empirically, the checkpoints at epoch 6 were used in testing. All experiments were run on Nvidia A-100 GPU clusters. With $K_{\text{train}} = 5$, the RA-VQA training takes around 5 hours (10 epochs) while testing takes 5 minutes. The time cost increases as K_{train} increases, approximately linearly.

Pre-trained model parameters (e.g. T5-large and BERT-base) are provided by `huggingface` [322] accompanied by Python libraries (under Apache License 2.0). FAISS [138] is under MIT License.

3.4.4 Baseline Systems

Retrieval Augmented Generation

RAG [167] is based on DPR and an answer generator that are trained jointly by approximately[4] marginalising the probability of y over the retrieved documents. In the notation of Sec. 3.3:

$$p_{RAG}(y|x) \approx \sum_{k=1}^{K} p_\phi(y|x,z_k)\ p_\theta(z_k|x). \tag{3.14}$$

The answer generator and the retriever are jointly trained by optimizing the RAG loss: $-\sum_{(x,\mathscr{S})\in\mathscr{T}} \log\left(p_{RAG}(s^*|x)\right)$. The rationale is that $p_\theta(z_k|x)$ will increase if z_k has a positive impact on answer generation [167]. We consider RAG as an important baseline and have carefully replicated its published implementation.[5]

Comparisons to Systems in the Literature

We compare against the published OK-VQA results from systems described in Sec. 3.2: **ConceptBERT**, **KRISP**, **MAVEx**, and **VRR**. We also report performance against unpublished (non peer-reviewed at the time of this study) systems **TRiG**, **PICa**, and **KAT**. **TRiG** uses a similar image-to-text transform as this work, so to enable fair comparison with our model we replicate their knowledge fusing method with our features. The results of these systems are reported in Table 3.1; the results marked * are our own. **TRiG***, our own implementation of TRiG, concatenates K encoder outputs for the decoder to use in generation.

We make some particular observations. Our TRiG* improves over the results released in its paper (VQA Score of 48.32 vs 45.51) at $K_{\text{train}} = K_{\text{test}} = 5$; TRiG and TRiG Ensemble both benefit from more retrieved documents in training and testing ($K_{\text{train}} = K_{\text{test}} = 100$), although at great computational cost. Best performance with KAT-T5 and VRR similarly requires large document collections in training and in test.

We include results from GPT-3 based systems because they are amongst the best in the literature, but we note that GPT-3 is so much bigger than T5 (175 billion parameters in GPT-3 v.s. 770 million in T5-large) that simply switching a system implementation from T5 to GPT-3 can give significant improvements: KAT-T5 achieved a 44.25 VQA Score while ensembling it with GPT-3 yields 54.41; and GPT-3 alone already achieved good performance with prompting (PICa with 48.00 VQA Score). Our RA-VQA system is based on T5, but we

[4] because we sum over the top-K documents instead of all, assuming the probabilities of the rest are small and irrelevant.

[5] The authors released RAG in `huggingface` [322]: https://github.com/huggingface/transformers/tree/master/examples/research_projects/rag

Model	T5	GPT-3	K_{train}	K_{test}	Knowl. Src.	PRRecall	HSR / FSR	H/F	EM	VQA
ConceptBERT	×	×	-	-	C					33.66
KRISP	×	×	-	-	C + W					38.35
VRR	×	×	100	100	GS					45.08
MAVEx	×	×	-	-	W + C + GI					39.40
KAT-T5	✓	×	40	40	W					44.25
TRiG	✓	×	5	5	W				49.21	45.51
TRiG	✓	×	100	100	W				53.59	49.35
TRiG-Ensemble	✓	×	100	100	W				54.73	50.50
TRiG*	✓	×	5	5	GS				52.79	48.32
RAG*	✓	×	5	5	GS	82.34	12.28 / 40.24	0.31	52.52	48.22
RA-VQA (Ours)	✓	×	5	5	GS	82.84	16.75 / 41.97	0.40	58.72	53.81
RA-VQA (Ours)	✓	×	5	50	GS	96.55	17.32 / 42.09	**0.41**	**59.41**	**54.48**
Ablation Study										
RA-VQA-FrDPR	✓	×	5	5	GS	81.25	15.01 / 40.76	0.37	55.77	51.22
RA-VQA-NoPR	✓	×	5	5	GS	77.67	15.97 / 41.83	0.38	57.80	52.98
RA-VQA-NoCT	✓	×	5	5	GS	83.77	14.55 / 42.96	0.33	57.51	52.67
RA-VQA-FrDPR-NoCT	✓	×	5	5	GS	81.25	13.18 / 41.82	0.31	54.99	50.66
GPT-3-based Systems (>175 Billion Parameters)										
PICa	×	✓	-	-	GPT-3					48.00
KAT-Knowledge-T5	✓	✓	40	40	W + GPT-3					51.97
KAT-Ensemble	✓	✓	40	40	W + GPT-3					54.41

Table 3.1 RA-VQA vs. systems in the literature. Ablation study is also incorporated. Knowledge Sources: **C**onceptNet; **W**ikipedia; **G**oogle **S**earch; **G**oogle **I**mages; **GPT-3** closed book knowledge. H/F: HSR to FSR ratio. PRRecall, HSR, FSR, and EM are reported in percentage (%). PRRecall is reported at the corresponding K_{test}.

still find competitive results even in comparison to systems incorporating GPT-3 (54.48 vs 54.41 of KAT-Ensemble).

3.4.5 RA-VQA Performance Analysis

We find that RA-VQA matches or improves over all baseline systems with a VQA Score of 54.48. This is with a configuration of $K_{\text{train}} = 5$ and $K_{\text{test}} = 50$, thus validating our claim that RA-VQA can use a large number of retrieved documents in testing (50) while using relatively few retrieved documents in training (5). We find that reducing the number of retrieved documents in test ($K_{\text{test}} = 5$) reduces the VQA Score, but still yields performance better than all baselines except the KAT ensemble.

We also find that RA-VQA performs well relative to GPT-3 baselines. RA-VQA yields a higher VQA score than KAT-Knowledge-T5 (54.48 vs. 51.97) and matches the KAT-Ensemble system. We emphasise that RA-VQA is significantly smaller in terms of parameters (and in model pre-training data) than these GPT-3 based systems and that training RA-VQA requires much less memory ($K_{\text{train}} = 5$ vs $K_{\text{train}} = 40$).

Contributions of Query Features and DPR to Overall Performance

Model	Q	O	A	C	T	VQA Score
RA-VQA-NoDPR	✓	×	×	×	×	28.05
RA-VQA-NoDPR	✓	✓	×	×	×	40.95
RA-VQA-NoDPR	✓	✓	✓	×	×	42.14
RA-VQA-NoDPR	✓	✓	✓	✓	×	45.31
RA-VQA-NoDPR	✓	✓	✓	✓	✓	46.16
RA-VQA-FrDPR	✓	✓	✓	✓	✓	51.22
RA-VQA	✓	✓	✓	✓	✓	53.81

Table 3.2 Ablation study on input features and system configurations: **Q**uestions; **O**bjects; **A**ttributes associated with objects; **C**aptions; visible **T**ext from OCR. $K = 5$ in RA-VQA and RA-VQA-FrDPR.

A detailed ablation study on input features is presented in Table 3.2. As shown, the T5 model fine-tuned on OK-VQA achieves a 28.05 VQA Score. The VQA Score increases to 46.16 as objects, object attributes, image captions, and OCR texts are incorporated into RA-VQA-NoDPR. With 5 retrieved documents, RA-VQA-FrDPR yields a 51.22 VQA Score, with a further improvement (53.81 VQA Score) in full training of retrieval and answer generation, confirming that outside knowledge is needed to answer OK-VQA questions.

From the feature ablation study we found that our RA-VQA-NoDPR achieved $\sim$28 VQA Score relying on only questions. This is due to the fact that $\sim 75\%$ of answers to training questions appear in the answers to test questions. We conduct a sanity check to ensure that this score is from random guesses and the dataset is not overwhelmingly easy.

As shown in Table 3.3, for each distinct question, the model learned to generate the same answer without access to the associated images. These random guesses can match to the answers of some test questions by chance, leading to a good VQA Score. By inspection we report that most of the successful cases are random guesses, and these questions are still not directly answerable without reading the associated images.

Benefits of Integrated Training

Joint training is a key benefit of our proposed RA-VQA framework: model predictions combine with pseudo relevance labels to improve retrieval, and the resulting improved retrieval in turn provides customised answer generation targets. To quantify these effects, we take RA-VQA-FrDPR as a starting point (Table 3.1). Comparing it with other RA-VQA models suggests that DPR training in itself is necessary, as using only pre-trained DPR (RA-VQA-FrDPR) leads to weaker VQA Score (51.22). Using model predictions alone

Question	Prediction
What type of bird is this?	hawk
What time of day is it?	afternoon
What breed of dog is this?	lab
What kind of dog is this?	chihuaha
What kind of bird is this?	hawk
What sport is this?	horse race
What breed of horse is that?	clydesdale
What century is this?	19th
What kind of birds are these?	pigeon
What food does this animal eat?	cat food
What city is this?	new york
What is the weather like?	rainy
What do these animals eat?	grass
What activity is this?	skateboard
How long do these animals live?	20 years
What type of train is this?	passenger
What is this used for?	travel
What is this room used for?	sleep
What kind of bird is that?	hawk
What kind of cat is this?	domestic
What food does the animal eat?	cat food
What type of dog is this?	chihuaha
What food do these animals eat?	cat food
What place is this?	switzerland
What breed of cat is this?	calico
What season is this?	winter

Table 3.3 Example of random guesses with only question input. Random guess achieved a good VQA Score by matching to the answers by chance. But the OK-VQA questions are still not directly answerable without access to the associated images.

in joint DPR training (RA-VQA-NoPR) leads to a higher VQA Score (52.98 vs 51.22), but a significantly lower PRRecall (77.67% vs 81.25%). The model decides to remove some pseudo relevant documents but achieves better performance. This points to a potential problem that can arise. Pseudo relevance is only an imperfect indication of true relevance and so is not an ideal criteria on its own. Training DPR to retrieve pseudo relevant documents could result in misleading documents being used in answer generation.

Using both pseudo relevance labels and model predictions in DPR training (RA-VQA) improves VQA Score to 53.81 and notably improves PRRecall to 82.84%. Including pseudo relevance ensures that potentially useful documents are retained, even while the generator is still learning to use them.

As noted, RA-VQA improves retrieval with the feedback of model predictions, and in turn the improved retrieval leads to improved answer generation by training towards s_k^*, a customised generation target for each retrieved document z_k. We remove this interaction from RA-VQA models by enforcing $s_k^* = s^*$ (the most popular human response), independent of the retrieved z_k. The ablated models are denoted with a `*-NoCT` suffix.

As shown in Table 3.1, customizing generation targets for each retrieved z_k in training yields performance boost for both RA-VQA-FrDPR and RA-VQA, showing that this supervision signal is beneficial to overall system performance. When generation targets are not customised for each retrieved document (RA-VQA-NoCT), VQA Score drops by 1.14 relative to RA-VQA, showing that customised generation targets play an important role in the overall system: by training the model to extract the reliable answers available in retrieved documents, answer generation and retrieval are both improved.

We also notice that the improvement brought by customised targets is larger for RA-VQA (+1.14 VQA Score) compared to RA-VQA-FrDPR (+0.56 VQA Score), showing that customizing the generation target brings more benefits when the retrieval is improved within our proposed RA-VQA joint training framework. This further confirms that the two components, retrieval and answer generation, complement each other.

Interaction of Retrieval and Generation

Table 3.1 also reports our investigation into the interaction between document retrieval and answer generation. In comparing RA-VQA-FrDPR (frozen DPR) to RA-VQA, we see that joint training of DPR yields not only improved EM but also significantly higher HSR (+1.74%) and FSR (+1.21%). Manual inspection of OK-VQA reveals that there are many general knowledge questions. For example, document retrieval is not needed to answer the question "Is this television working?" in reference to a picture of a broken television lying in a field. A high FSR indicates good performance on such questions. By contrast, a high HSR reflects the ability to use document retrieval to answer the questions that truly require external documents.

Both EM and HSR are further improved for $K_{\text{test}} = 50$ in RA-VQA, with little change in FSR. The increased HSR to FSR ratio (0.41 vs. 0.40) indicates that RA-VQA is using these additional retrieved documents to answer the questions that need outside knowledge.

HSR and FSR also explain the relatively weak performance of RAG*. We see that although RAG* and RA-VQA-FrDPR have similar FSRs, RAG* has higher PRRecall but lower HSR (by -2.73%). This suggests RAG*'s DPR model is not well matched to its answer generator. The result is that retrieved documents remain unexploited. In manual examination of gradients of document scores in training, we find anecdotally that adjustments to document

scores are often counter-intuitive: documents that do not contain answers can still have their scores upvoted if the answer generator happens to find a correct answer by relying only on the ability of T5 model. This works against a model's ability to find answers in retrieved documents even when those documents are relevant.

Effects of K_{train}

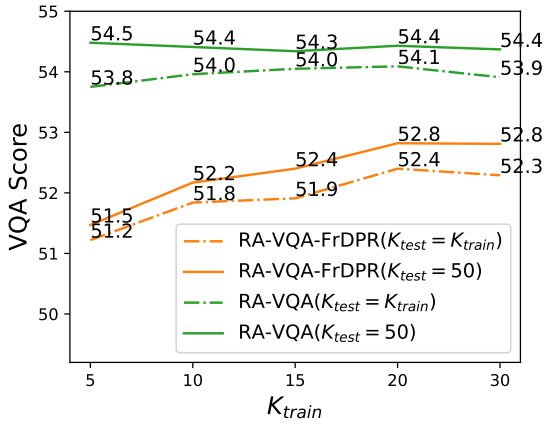


Fig. 3.4 VQA Scores against K_{train}. Dashed line: $K_{\text{test}} = K_{\text{train}}$; solid line: $K_{\text{test}} = 50$. Our proposed model achieves the best performance when additional documents are retrieved in test ($K_{\text{test}} = 50$). This holds even for models trained to retrieve fewer documents.

As noted, retrieving a large collection of documents in training is costly (large K_{train}). Fig. 3.4 shows that RA-VQA can be trained with relatively few retrieved documents ($K_{\text{train}} = 5$). We gradually increase K_{train} while fixing $K_{\text{test}} = K_{\text{train}}$ (dash lines) and $K_{\text{test}} = 50$ (solid lines). RA-VQA achieves consistent performance ($\sim$54.4 VQA Score) at $K_{\text{test}} = 50$, which suggests that our joint training scheme is able to gather most useful knowledge into a top-50 list even when the model is trained to retrieve fewer documents. This is not the case for the frozen DPR systems which require increasing K_{train} to obtain best performance. RA-VQA's superior performance shows that joint training of retrieval and generation yields clear benefits in computation and answer quality.

Further Evaluating the Retrieval Performance of RA-VQA

In addition to Pseudo Relevance Recall (PRRecall) introduced above, we further evaluate retrieval performance with **Pseudo Relevance Precision (PRPrec)@K**, which is calculated

as the rate of pseudo positive documents in all the K documents retrieved for a question:

$$\text{PRPrec@K} = \frac{1}{K}\sum_{k=1}^{K} H(z_k, \mathscr{S}) \tag{3.15}$$

where $H(\cdot)$ is the pseudo relevance function introduced in Sec. 3.3.2.

Models	$K=5$		$K=10$		$K=20$		$K=50$		$K_{\text{test}}=5$	
	P	R	P	R	P	R	P	R	EM	VQA Score
VRR [208]	-	80.40	-	88.55	-	93.22	-	**97.11**	-	42.54
RA-VQA-FrDPR	51.82	81.25	49.20	88.51	45.98	92.98	41.24	96.75	55.77	51.22
RA-VQA	**57.39**	**82.84**	**54.83**	**89.00**	**51.48**	**93.62**	**46.36**	96.47	**58.72**	**53.81**

Table 3.4 Comparing retrieval performance of VRR and our RA-VQA models. The same knowledge corpus (`GS-full`) was used. P: Pseudo Relevance Precision; R: Pseudo Relevance Recall; EM: Exact Match. P under $K=5$ refers to PRPrec@5. VRR was trained on $K_{\text{train}}=100$, while RA-VQAs were trained on $K_{\text{train}}=5$.

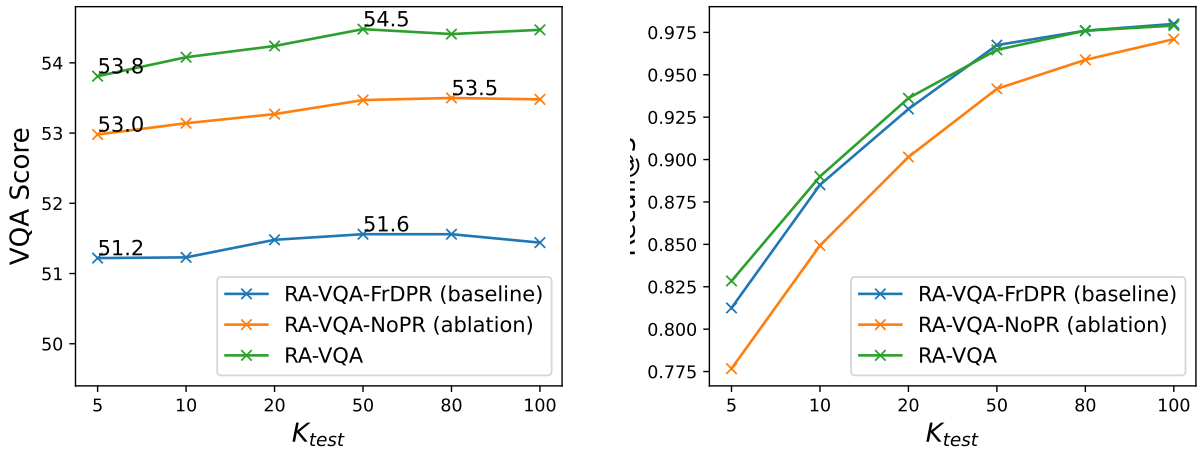
The success of our RA-VQA model can be further explained by Table 3.4. As expected, RA-VQA-FrDPR (pre-trained DPR) achieves similar retrieval performance as VRR [208] since they are both based on DPR and are trained with the same pseudo-relevance-based labels. Our proposed RA-VQA, with a substantial improvement in Recall over RA-VQA-FrDPR (82.84 PRRecall@5 vs 81.25 PRRecall@5), achieves significantly higher Precision (57.39 PRPrec@5 vs 51.82 PRPrec@5). This also yields substantial improvements to both EM (+3.05%) and VQA Score (+2.59%). This suggests that training the retriever jointly presents more potentially relevant documents to answer generation, improving the quality of the top-ranked documents.

Effects of Retrieving More Documents in Test

Fig. 3.5 presents the change of VQA Score and PRRecall as additional documents are retrieved in test (increasing K_{test}).

PRRecall is improved dramatically as K_{test} increases from 5 to 50, after which only marginal improvement is observed. Similarly, the VQA Score of these models is improved as more documents are presented in test, and the performance peaks at $K_{\text{test}} \sim 50$. This suggests that including more additional documents in test is more likely to include the truly relevant document to help answer the question yet along with more distracting and misleading documents.

RA-VQA-NoPR, which uses only model predictions in training to adjust document scores without pseudo relevance labels, yields a significantly lower PRRecall curve (orange curve



(a) VQA Score vs K_{test}

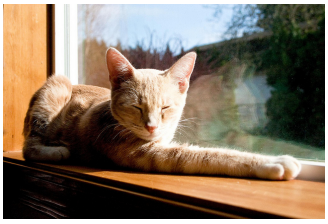
(b) Recall vs K_{test}

Fig. 3.5 Comparison of model performance as more documents are retrieved in testing. These models are all trained with $K_{\text{train}} = 5$. In RA-VQA full joint training (green), combining model predictions with pseudo relevance labels yields higher PRRecall at low K_{test}, showing that full joint training improves retrieval; RA-VQA-NoPR (orange), which uses only model predictions in training, achieves a higher VQA Score with lower Pseudo Relevance Recall compared to the RA-VQA-FrDPR with frozen DPR in training (blue), which suggests that Pseudo Relevance is only an approximate measurement of actual relevance.

in Fig. 3.5b) than RA-VQA-FrDPR (blue curve) across all K_{test}s, but achieves much higher VQA performance (Fig. 3.5a). This further confirms that Pseudo Relevance Labels are a weak signal and a high PRRecall does not necessarily guarantee to gather truly relevant knowledge in retrieval.

Example System Outputs

To answer the question of how our proposed framework improves retrieval and answer generation, we qualitatively evaluate the outputs of our framework. Since the document being used to provide the final answer is explicit in our framework, we are able to compare the retrieval results and the generated answer by examining the system's outputs. We present some example system outputs in Fig. 3.6 to compare RA-VQA-FrDPR and our proposed RA-VQA framework. Insights and conclusions are provided through the red and green boxes in Fig. 3.6.



How many teeth does this animal use to have?

RAVQA-Frozen

an adult dog should have 42 teeth in total: that's 20 on top of their jaw and 22 on the bottom. → Pred: 42 ✗

cats have 30 teeth and dogs have 42. → Pred: 30

we humans have 32 pearly whites. horses have 44 chompers, dolphins can have as many as 250 teeth and, it's hard to believe, but snails can have more than 20,000 tiny, very sharp teeth — located on their tongues → Pred: 32

RAVQA (ours)

cats have 26 deciduous teeth and 30 permanent teeth. → Pred: 26 ✓

this is true. as dogs grow older, they will have 42 permanent teeth while cats will have 30. → Pred: 30

cats have 26 deciduous teeth consisting of: 12 incisors; 4 canines; 10 premolars. permanent teeth begin to erupt from the age of 11-12 weeks. → Pred: 26

Retrieval was improved by RA-VQA, which leads to sucessful answering with given knowledge.

What is the active ingredient in this?

RAVQA-Frozen

i credit this “secret ingredient” for being the greatest offender in this recipe. → Pred: toothpaste

ingredients on the list that end in “ose”—fructose, maltose, sucrose..... → Pred: toothpaste

natural cheese is made from only four ingredients: milk, salt, starter culture (good bacteria) and rennet (an enzyme) → Pred: toothpaste ✗

Questions about "ingredient" are common in food domain, and misleading material may be presented by Pseudo Relevance Labels to Answer Generator, leading to failed answering.

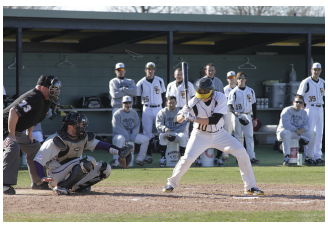
RAVQA (ours)

active ingredient sodium fluoride 0.21% (0.12% w/v fluoride ion) purpose anticavity toothpaste use helps protect teeth and roots against cavities warnings keep out of reach of children under 6 years of age. → Pred: fluoride ✓

ingredients. active ingredient - purpose. sodium fluoride (0. 24%) - anticavity toothpaste. inactive ingredients: sorbitol, water, hydrated silica, peg-32, sodium lauryl sulfate, ... → Pred: fluoride

fluoride-containing compounds in the form of sodium monofluorophosphate, sodium fluoride and stannous fluoride are used as anticaries agents in toothpastes. → Pred: toothpaste

RA-VQA sucessfully retrieved more relevant documents.



What position does the man with the bat play?

RAVQA-Frozen

catcher is a position for a baseball or softball player. when a batter takes their turn to hit, the catcher crouches behind home plate, in front of the (home) umpire, and receives the ball from the pitcher..... → Pred: Catcher ✗

.....

RAVQA (ours)

catcher is a position for a baseball or softball player. when a batter takes their turn to hit, the catcher crouches behind home plate, in front of the (home) umpire, and receives the ball from the pitcher..... → Pred: Batter ✓

.....

Even with the same retrieved document, RA-VQA learned to retrieve correct answers.

Notations

Pred: answer — Prediction (correct / wrong) with the given knowledge

Pred: answer — Selected as final prediction

Fig. 3.6 Example system outputs comparing RA-VQA-FrDPR (baseline) and our RA-VQA that benefits from joint training of retrieval and answer generation.

Generalizing RA-VQA to Other Datasets

We are also interested in whether this approach is also generalisable to other similar VQA tasks that may benefit from improved passage retrieval.

We implement our framework on another knowledge-based VQA task, Fact-based VQA (FVQA) (introduced in Sec. 2.4.2). This dataset contains commonsense factoid VQA questions, such as "Question: which object in the image can cut you? Answer: the knife". In contrast to OK-VQA where no knowledge base is provided, FVQA grounds each question-answer pair with a fact (a triplet from several 'common sense' knowledge bases, including ConceptNet [288], Webchild [295], and DBpedia [13]). A triplet contains a head node, a relation, and a tail node (e.g. [Car] `/r/HasA` [4 wheels]). To cope with passage retrieval, these knowledge triplets are flattened into surface texts (e.g. "[car] has [4 wheels]") such that DPR can be directly applied to retrieve them. We replace pseudo relevance with ground-truth relevance since relevant triplets for answering questions are given.

The metrics used for assessing performance are Accuracy and Recall, with their standard deviations of 5 splits. Accuracy counts the portion of questions that are successfully answered, while Recall@K measures how likely the retrieved K knowledge triplets contain the answer node. Since FVQA was designed for answer selection instead of open-ended answer generation, prior works used accuracy as "whether the answer node is successfully selected from all KG nodes". To enable fair evaluation with our open-ended framework, in calculating accuracy, a question is considered successfully answered if the answer node is the closest node to the generated answer string (shortest in Levenshtein distance). For example, the generated answer 'knives' is still a valid answer since the answer node '[knife]' can be matched with a shortest Levenshtein distance.

The significance of performance is guaranteed by reporting the average of 5 splits (as in the official FVQA evaluation). In total we trained 5 DPR models and 5 $\times$ 3 models (RA-VQA, RA-VQA-FrDPR, and RA-VQA-NoDPR) with the same hyperparameters. Each split has approximately half questions for training and the remaining for testing.

We compare with three systems in prior work:

(1) FVQA [311]: the baseline system provided in the official FVQA dataset paper.

(2) GCN [226]: a model that leverages graph convolutional networks (GCNs) to aggregate features from visual/language/fact modalities.

(3) Mucko [390]: the state-of-the-art system (up to the time of this study) that uses GCNs to combine visual, fact, and semantic graphs.

As shown in Table 3.5, RA-VQA-NoDPR achieves an already strong result (67.93% accuracy) compared to early work in FVQA, showing that the extracted vision-to-language features are useful and text-based Transformers are able to learn to answer commonsense

Model	Accuracy (Std.)	Recall@5 (Std.)
Mucko (Zhu et al. [390])	73.06 (–)	-
RA-VQA (ours)	69.88 (0.13)	68.77 (0.87)
GCN (Narasimhan et al. [226])	69.35 (–)	-
RA-VQA-FrDPR (ours)	68.81 (0.59)	64.54 (0.80)
RA-VQA-NoDPR (ours)	67.93 (0.82)	-
FVQA (Wang et al. [311])	58.76 (0.92)	-

Table 3.5 Model performance on the FVQA dataset (sorted by accuracy). Our proposed systems are in bold.

VQA questions well without accessing the provided knowledge graph (ConceptNet). The incorporation of DPR boosts the performance to 68.81% with 64.54% Recall@5, showing that retrieval works as expected and the retrieved knowledge triplets are exploited in answer generation. The joint training scheme improves the retrieval (64.54% to 68.77% Recall@5) as well as the overall performance (68.81% to 69.88% Accuracy). This demonstrates that our proposed joint training framework is generalisable to other KB-VQA tasks, though the passages used in retrieval are simply flattened surface texts of KG triplets.

In comparing with other systems in the FVQA benchmark, our best system ranks second without an explicit design for leveraging KG structures. This shows the power of open-ended answer generation with text-based Transformers. But we emphasise that better performance could be achieved through designing a more specialised retrieval component for the structured knowledge base used in this task.

To summarise, our system shows great generalisability in an external KB-VQA task that was constructed very differently. Therefore, the proposed framework can serve as a strong basis for future improvements.

3.5 Limitations and Future Work

One possible limitation is that some relevant information (such as relative positioning of objects in the image) could be lost in transforming images independently of the information being sought. Extracting visual features based on queries could be a natural next step, although query-specific processing of the image collection would be computationally expensive. We investigate models capable of handling both visual and textual input in later chapters (Chapter 5 and 6).

We selected the Google Search corpus [208] as the knowledge base for our question answering system. Its advantages are that it is large, openly available, and can be readily

used to replicate the results. However some visual question types (e.g. 'Is the athlete right or left handed?') could plausibly require both complex reasoning and more closely relevant documents from additional knowledge sources (such as Wikipedia). We will experiment with other knowledge sources in Chapter 5.

3.6 Summary

In this chapter, we introduced Retrieval Augmented Visual Question Answering as a novel modelling framework for integrated training of DPR and answer generation. We have evaluated RA-VQA on the OK-VQA task and we find significantly better performance than the independent training of component system. Through diagnostic metrics such as HSR and FSR we analysed the interaction between retrieval and generation, and have also shown how RA-VQA's gains arise relative to other approaches, such as RAG. As a further practical benefit, we found that RA-VQA can be used with larger numbers of retrieved documents than were used in system training, yielding computational savings without sacrificing performance. We further evaluated the system qualitatively by examining the system outputs. The proposed framework was tested on another KB-VQA dataset, FVQA, and showed excellent performance. Therefore, we can conclude that RA-VQA is a generalisable and powerful framework for Knowledge-based VQA with outside knowledge.

Additionally, the work discussed in this chapter addresses research questions RQ1, RQ2, and RQ3, details of which will be elaborated in the final chapter of the thesis (Sec. 9.1).

In the next chapter, we will look into the limitations of current KB-VQA datasets, and propose a new dataset, Fact-based Visual Question Answering 2.0 (FVQA 2.0) to address them.

Chapter 4

Introducing Adversarial Samples into Fact-based Visual Question Answering

4.1 Introduction

In the preceding chapter, we introduced the RA-VQA framework, which achieved very strong performance on the Fact-based Visual Question Answering (FVQA) [311] task. This chapter examines the limitations inherent in this task, thereby identifying the challenges and exploring potential solutions to these issues.

FVQA is a KB-VQA task in which visually-grounded questions and answers about images are grounded by knowledge-graph (KG) triplets taken from several 'common sense' knowledge bases, such as ConceptNet [288], Webchild [295], and DBpedia [13]. An example is that "Question: Which thing in the image can be used for scooping food? Answer: spoon" is associated with the KG triplet "spoon - UsedFor - scooping food". These questions are challenging in that retrieving information from external KGs is necessary.

The original FVQA dataset [311] has several readily observed limitations. First, the dataset is small (5486 samples) and the annotations are limited to a single answer per question, ignoring other correct answers. This limitation arises from the FVQA creation process in which annotators were first asked to select a KG triplet on which they would ask a question about an image. This approach prevented the annotators from labeling other valid KG triplets. Secondly, the dataset is highly imbalanced. Some triplets and answers are frequently used, but other KG triplets and answers are severely underrepresented in training. For example, there are 1,129 possible answers in total, but over 90% of questions focus on only a half of them; 792 (70%) of answers appear less than 3 times; only 4,216 out of $\sim$220k triplets are used.

These limitations lead to a potential problem: KB-VQA systems trained on this dataset overfit on these frequently used triplets and perform poorly on other valid triplets or other images. Also, extensive overlap between training and test can lead to an unrealistically high question answering baseline performance. We noted that a question with a triplet unseen in training is often answered with 'person', since it is the most frequent answer in the original data distribution.

To overcome these limitations, we introduce an enlarged test set that contains two types of adversarial samples (as shown in Fig. 4.1): (1) *FixQ*: the question remains the same, but is

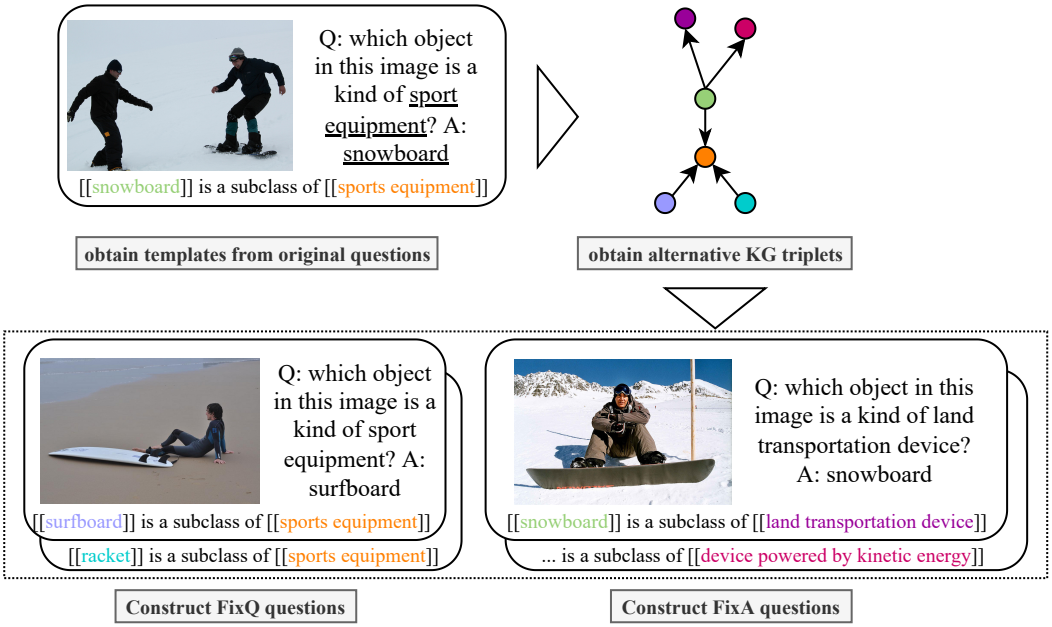


Fig. 4.1 The workflow of constructing adversarial samples (FixQ and FixA questions) from the original test set questions.

associated with a different image and a different correct answer. This ensures that a system is less able to achieve high performance if it is biased by language patterns in questions; (2) *FixA*: the answer remains the same, but the question is asked in a different way. This favours systems that do more than make straightforward associations between questions and answers based on the training data. In contrast to the original test set, this new set further challenges KB-VQA system to retrieve knowledge from KBs and answer questions without being biased towards frequent answers in the original dataset. We show that models trained on the original FVQA training sets are significantly less robust on these adversarial test samples.

Given that it is hard to guarantee a good triplet coverage during annotation, we explore an augmentation scheme to address this problem without costly human annotation of large-scale

adversarial training samples. Our scheme generates slightly noisy adversarial samples that improve the coverage of valid KG triplets to enhance model training.

In this chapter, our contributions are:

(1) We introduce FVQA 2.0, which adds an adversarial test set that challenges KB-VQA system robustness to adversarial variants of questions.

(2) We demonstrate the performance gap between the original test set and the adversarial test set, showing that considering adversarial samples is important for better realistic KB-VQA performance.

(3) To further demonstrate the importance of adversarial samples, we leverage a semi-automated augmentation scheme to improve system robustness on the adversarial test through the creation of large-scale noisy adversarial examples.

The data have been released at:

https://github.com/LinWeizheDragon/Retrieval-Augmented-Visual-Question-Answering/tree/legacy_v1.

4.2 Related Work

This section provides an overview of the related work up to October 2022, the time of this study.

As discussed in Sec. 2.4.2, KB-VQA questions can focus on facts and concepts, as in FVQA [311] and OK-VQA [214]. Such questions challenge the information retrieval ability of systems. KB-VQA questions can also require commonsense reasoning, as in parts of OK-VQA and A-OKVQA [273]. In particular, S3VQA [126] is an augmented version of OK-VQA, improving both the quantity and quality of some question types. A-OKVQA has shifted its core task to 'reasoning questions'. Only 18% of questions in A-OKVQA require answers from an external knowledge base.

VQA 2.0 [95] collects 'complementary images' such that each question is associated with a pair of images that result in different answers. Jain et al. [126] derive new S3VQA questions from manually defined question templates. They annotated spans of objects that could be replaced, and then substituted them with a complicated substitute-and-search system. In contrast to their labour-intensive annotation work, our adversarial samples are collected through a semi-automatic approach that fully leverages the structural information in KGs to significantly reduce the human work required.

More broadly, in Knowledge-Graph Question Answering (KG-QA), work has exploited KG to generate synthetic data in unseen domains [192, 305, 193]. Our work extends visually-grounded questions with valid common sense KG triplets.

4.3 Method

4.3.1 Extracting Question Templates

We extract question templates that can be used to reconstruct new questions using other valid KG triplets. We apply a rule-based system to replace KG entities that appear in the questions. For example, 'what is used for storing liquid in this image?' is transformed to 'what is used for <t> in this image?' given that the associated KG triplet is "bottle (<h>) - /r/UsedFor (<r>) - store liquid (<t>)".

For each template, we construct new question-answer pairs by exploring the node structure of the KG. For example, "bottle - /r/UsedFor - hold water" is also a valid triplet from ConceptNet, whose head and relation are the same as the original triplet. A new question "Q: what is used for holding water in this image? A: bottle" can now be constructed.

4.3.2 Template Filtering

We focus on questions about object concepts that are transferable to other images, ignoring a small portion ($<10\%$) of FVQA questions to which the answers are based on particular scenes (e.g. 'what can you often find in the place shown in this picture?').

Human annotators are employed to filter out non-transferable templates, such as questions that contain specific object positioning ('what is the object in the lower right of this image used for?'). This process takes around 1 hour with two annotators to obtain 440 valid templates after removing highly similar templates.

4.3.3 Matching Suitable Images

We use 619 of FVQA images[1] that are also present in the Visual Genome dataset [153]. Using the object annotations of the VG dataset to determine if an image contains the object being asked, we employ a rule-based system to assign a suitable image to each generated adversarial sample.

When assigning suitable images to question templates, it is important to ensure the diversity of images being used. This is achieved by regulating the number of assignments per image through a straightforward approach, ensuring the distribution is approximately equal across all images: in this process, for each new question-answer pair requiring an image, all images containing the object mentioned in the question are ranked by their current total number of assignments. The image that meets the criteria and has the fewest assignments

[1] FVQA images are from Microsoft COCO [181] and ILSVRC [258].

is selected as the associated image for the new sample. By implementing this simple yet effective strategy, we found that the assigned images exhibit good diversity.

We limit the number of FixQ and FixA questions generated by each template to 5, which guarantees a reasonable dataset size. 3,805 questions are generated.

4.3.4 Manual Verification

We conduct manual verification to rule out samples that are incorrectly generated. Two annotators (volunteers in the research group) worked independently to rule out incorrectly generated examples. An example was accepted only if the two annotators achieved consensus. The annotators attempted to fix grammar errors that caused severe misunderstanding, while mild errors were kept (for example, 'is used for carry people' does not prevent models/people from understanding the question, and thus the annotators are not required to fix them).

In particular, questions that might contain information of individuals/private information were dropped, though they are very rare cases.

When a question can be answered with multiple instances in an image, all possible answers are included. During annotation, incorrect answers were dropped from the list. In evaluation, answering any correct answer is considered successful. There are around 11% multiple-answer questions at the end of the verification process.

As a result, 432 counter-intuitive KG triplets are removed in this step. Finally, we obtain 2,820 adversarial samples, offering 1,671 new valid triplets from the KG. Around 75% samples are verified as correct, showing that the rule-based generation works well. The remainder are discarded.

The official FVQA evaluation performs 5-fold validation: each split preserves around half its samples for testing. As a naming convention, under each split, the templates extracted from the original training samples are called 'train templates' while the rest are 'test templates'. Since the train templates may contain language patterns that could be learned in training, we ensure that only questions derived from test templates are used in the adversarial testing. As a result, we have 1,376 adversarial test samples per split on average, with 1,129 FixQ and 246 FixA questions. The origins of these adversarial test samples are referred to as '*Originating Questions*'. There are 435 such questions per split on average.

4.3.5 Augmentation with Adversarial Data

We explore an augmentation scheme to augment the training data with slightly noisy but auto-generated adversarial samples, which avoids heavy annotation work. In each split, **only the train templates** (defined in the above paragraph) are used to generate adversarial

samples for training such that no information of test samples is leaked to training. This avoids biasing the training to the test sets, which would make the test sets less indicative of true system performance. We obtain an augmentation set with 2,262 questions per split on average semi-automatically, which would otherwise cost hundreds of hours to build from scratch. In training, the original question-answer pairs of these generated adversarial samples are randomly replaced by their adversarial variants.

4.4 FVQA 2.0

4.4.1 Dataset Statistics

Dataset Name	Set Name	#Samples	std
FVQA	Standard Train Set	2,927	69
	Standard Test Set	2,899	69
FVQA 2.0	Originating Questions Set	435	52
	Adversarial Test Set	1,376	193
	- FixA Questions	1,129	157
	- FixQ Questions	246	38
	Augmentation data	2,262	267

Table 4.1 Dataset statistics of the FVQA and FVQA 2.0 dataset. Standard Train Set/Test Set refer to the original FVQA dataset. #Samples: average number of samples across 5 folds; std: the standard deviation over 5 folds.

The numbers of samples in each set are provided in Table 4.1. The official FVQA dataset creates 5 folds by splitting the images being used. Half of these images are used in training while the other half are reserved for testing. In all our new sets, under each split, questions for training are not leaked to testing. The 'Originating Question Set' is a subset of Standard Test Set by its definition (Sec. 4.3.4). The Adversarial Test Set is formed by FixA questions and FixQ questions; it is created by generating adversarial question variants from the questions in the 'Originating Question Set' following the procedure described in Sec. 4.3. It covers relationships such as /r/RelatedTo, /r/IsA, /r/PartOf, /r/HasA, /r/UsedFor, /r/CapableOf, /r/AtLocation, /r/Desires, /r/MadeOf. The augmentation data consists of adversarial variants that are derived from the questions in the Standard Train Set.

4.4.2 Examples of FVQA 2.0

We demonstrate some examples from the new Adversarial Test Set in Fig. 4.2.

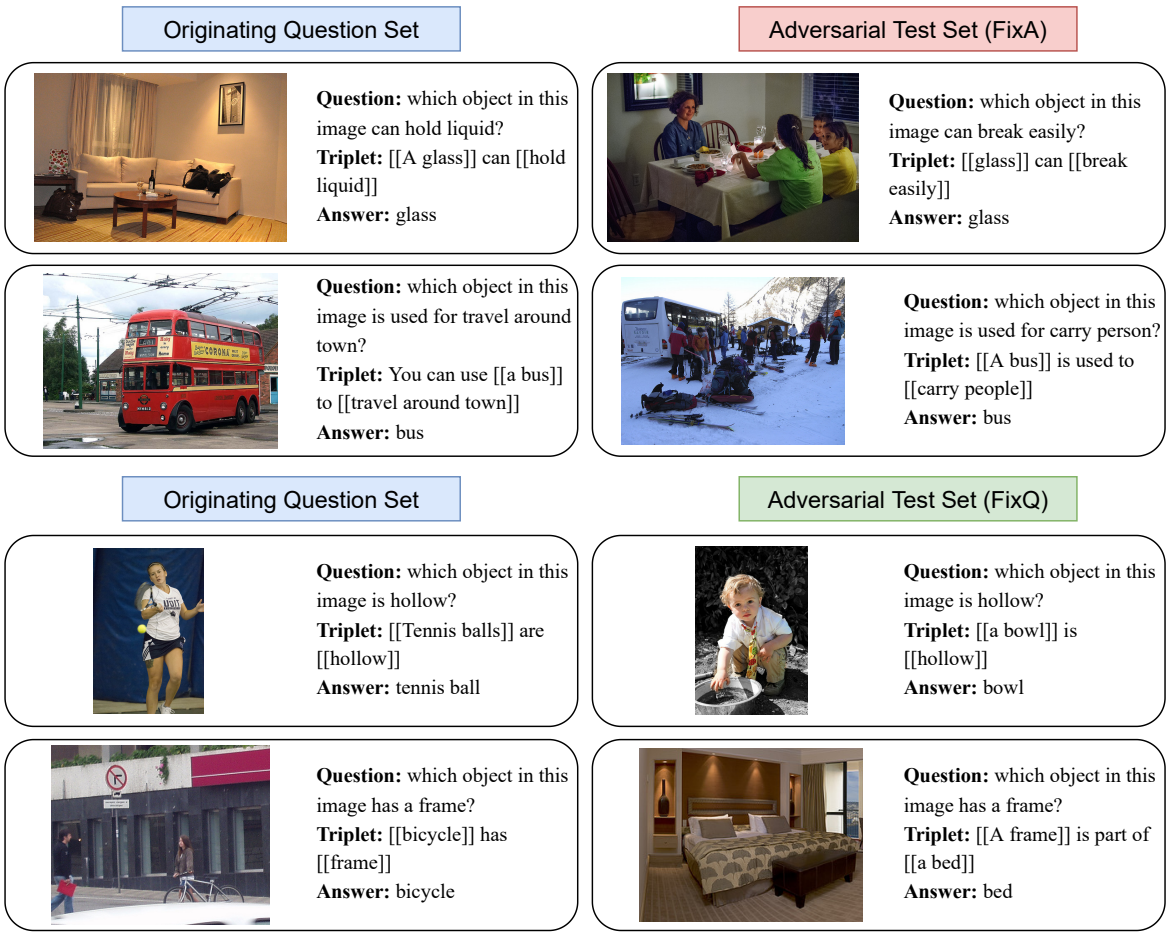


Fig. 4.2 Examples taken from the FVQA 2.0 adversarial test set. The questions in the left column are from the official FVQA test set. They are used to derive the adversarial questions in the right column. FixA: the answer remains the same while the way of asking for the answer is different; FixQ: the question remains the same, but the answer changes in a different image. More details are presented in Sec. 4.1.

4.5 Experiments

4.5.1 Systems for Comparison

We include the performance of several FVQA systems for comparison[2]: FVQA [311], the baseline system provided in the official FVQA dataset paper; GCN [226], a model that leverages graph convolutional networks (GCNs) to aggregate features from visual/language/fact modalities; Mucko [390], the state-of-the-art system as of date that uses GCNs to combine visual, fact, and semantic graphs.

[2]Since many FVQA systems are not open-sourced, we additionally include systems specialised for OK-VQA from Chapter 3.

We test our augmentation scheme on several systems that have code available: **RA-VQA-NoDPR** and **RA-VQA-DPR** [183], T5 [248]-based models that transform images into texts (e.g. objects, attributes, and image captions) and the DPR version additionally uses Dense Passage Retrieval [143] to retrieve documents from knowledge bases[3] (from Chapter 3); **TRiG** [88], a model that is similar to RA-VQA-DPR but differing in embedding fusion; **ZS-F-VQA** [47], an FVQA system that obtains the final prediction by fusing the individual predictions in answer/fact/relation graphs.

4.5.2 Metrics

We report accuracy and standard deviation over the 5 splits (Sec. 4.4.1). In calculating accuracy for open-ended generation systems (RA-VQA/TRiG), a question is considered successfully answered if the generated answer string is an exact match to the ground-truth answer node. We obtain the ground-truth answer node by computing the Levenshtein distance bewteen the ground-truth answer string and the name of each KG node in the dataset. The KG node that has shortest distance with the answer string is selected as the ground-truth answer node.

4.5.3 Training Details

ZS-F-VQA: The experiments were performed on 1 $\times$ Nvidia RTX 3090. We used the code from the official repository[4]. The original paper dropped questions that have rare answers. For fair comparison with other models, we added these rare answers back and performed training and testing. We chose to report the performance of the system which uses 'SAN' as the base model (details are in the paper and the repository), since this setting has achieved the best performance. The hyperparameters for training are kept the same as the original paper. In testing, we selected $k_e = 10; k_r = 1; score = 10$ by grid search (search range: $0 \leq k_e \leq 20; 0 \leq k_r \leq 20; 0 \leq score \leq 20$), with the objective of maximising the average accuracy over the 5 splits.

RA-VQA-NoDPR/RA-VQA-DPR/TRiG: All experiments were performed on 1 $\times$ Nvidia A-100 GPU. We chose Adam [149] as the optimizer. When the model has a DPR component, we trained the DPR component for 4 epochs with a constant learning rate 10^{-5}. In training the answer generator, the learning rate linearly decays from 6×10^{-5} to 0 after 10 epochs, as did in training the RA-VQA system in Chapter 3. For each split, the checkpoints at global step 2k (around 3.5 epochs) were used in testing. We retrieve 5 best documents

[3] In our experiments, the knowledge base consists of surface texts of triplets (e.g. "[car] has [4 wheels]").
[4] https://github.com/China-UK-ZSL/ZS-F-VQA

when predicting the answer ($K_{\text{train}} = 5$), since this number was reported to best balance the computation and performance in Chapter 3.

We obtained the pre-trained model parameters (T5-large and BERT-base) from `huggingface` [322]. These systems are implemented with `huggingface` Python libraries (under Apache License 2.0). The FAISS [138] system is under MIT License.

4.5.4 Performance and Discussion

Test on:	Standard Test Set		Originating Question Set		Adversarial Test Set	
Trained on:	Original	Augmented	Original#	Augmented (improv. over #)	Original*	Augmented (improv. over *)
ZS-F-VQA	48.16 ±1.03	48.57 ±1.00	63.67 ±0.88	64.63 ±0.81 +0.96	49.97 ±2.37	74.06 ±1.92 +24.09
TRiG	64.94 ±0.93	65.73 ±0.33	81.67 ±1.12	83.48 ±1.89 +1.81	68.86 ±3.26	79.79 ±1.34 +10.93
RA-VQA-NoDPR	66.19 ±1.15	66.70 ±1.00	84.59 ±1.24	85.75 ±0.90 +1.16	71.48 ±2.08	82.38 ±1.65 +10.90
RA-VQA-DPR	69.56 ±0.78	69.90 ±0.56	87.52 ±1.68	88.33 ±1.40 +0.81	76.91 ±1.93	85.05 ±1.15 +8.14

Table 4.2 Model performance on the standard test set, originating questions (from which the adversarial questions are derived), and adversarial test set. Results are reported as the average of 5 folds with standard deviations. The performance of three models that do not have code available: 58.76 (FVQA), 69.35 (GCN), and 73.06 (Mucko, state-of-the-art as of date) on the original Standard Test Set.

Table 4.2 shows that the systems used for evaluating the new adversarial set are sufficiently strong (e.g. 69.56% accuracy by RA-VQA-DPR) in comparison with the three models that do not have code available, which achieve 58.76% (FVQA), 69.35% (GCN), and 73.06% (Mucko, state-of-the-art as of date) respectively. RA-VQA-NoDPR achieves 84.59% accuracy on the originating questions but obtains only 71.48% accuracy on the adversarial samples derived from them. Such performance gaps are readily observed on all systems. Systems trained on the original training sets fail to perform equally well on the two sets, showing that the original FVQA training data does not contain adversarial variants and the resulting systems are vulnerable to them.

By incorporating adversarial variants in training, all systems achieve much better performance on the challenging adversarial set, e.g. RA-VQA-NoDPR is improved from 71.48% to 82.38% (+10.9%). The performance on the standard and adversarial test sets now match well, with the gap reduced from more than 10% to ∼3%, showing that the augmentation scheme significantly improves systems' reliability and robustness. The relative improvement is slightly less (+8.1%) for RA-VQA-DPR, which is expected given that it is a retrieval-based system designed to answer both seen and unseen questions with its strong retrieval ability. ZS-F-VQA benefits greatly from augmentation: its adversarial performance is improved

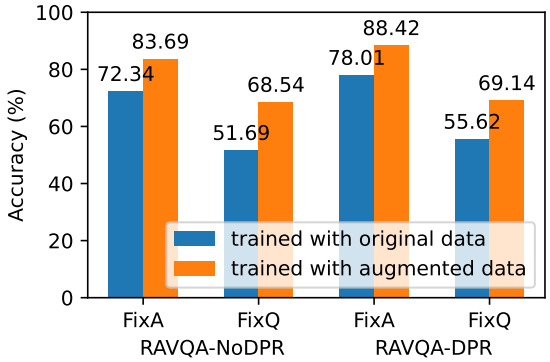


Fig. 4.3 Performance on FixQ and FixA questions.

by 24.09%. This is because its model size is much smaller and it can easily be biased by language patterns, images, and frequent answers seen in training.

The benefits derived from data augmentation are less pronounced on the Originating Question Set compared to the Adversarial Test Set (comparing the green numbers in Table 4.2). This suggests that merely increasing the amount of training data is not the sole factor driving significant improvements on adversarial test samples; rather, the inclusion of adversarial training samples plays a critical role in achieving these substantial gains.

In summary, systems trained on the original training sets are vulnerable to adversarial variants of the test questions. We show that through generating adversarial samples for data augmentation, systems become much more robust to these variants.

4.5.5 Analysis of Model Vulnerability

As shown in Fig. 4.3, RA-VQA systems trained with original training sets perform better on FixA questions ($\sim$88%) than on FixQ questions ($\sim$69%). This suggests that systems perform worse when asked the same questions on different images. This is potentially because the language patterns seen in training bias the models to frequent choices, lowering the FixQ generalizability. In contrast, systems are less distracted by different ways of asking for the same answer, potentially due to the strong language modelling capability of T5 used by them. The augmentation scheme improves systems on both types of questions significantly (by $\sim$10% on each), showing the value of adversarial samples in training.

Fig. 4.4 plots the RA-VQA-DPR performance on the adversarial test set questions that are grouped by their answer occurrences in the original FVQA dataset. The answer distribution of the original dataset affects adversarial performance greatly: systems perform much worse on questions whose answers appear less frequently in FVQA. In contrast, the performance deterioration that arises from answer rarity is mitigated significantly after augmentation. The

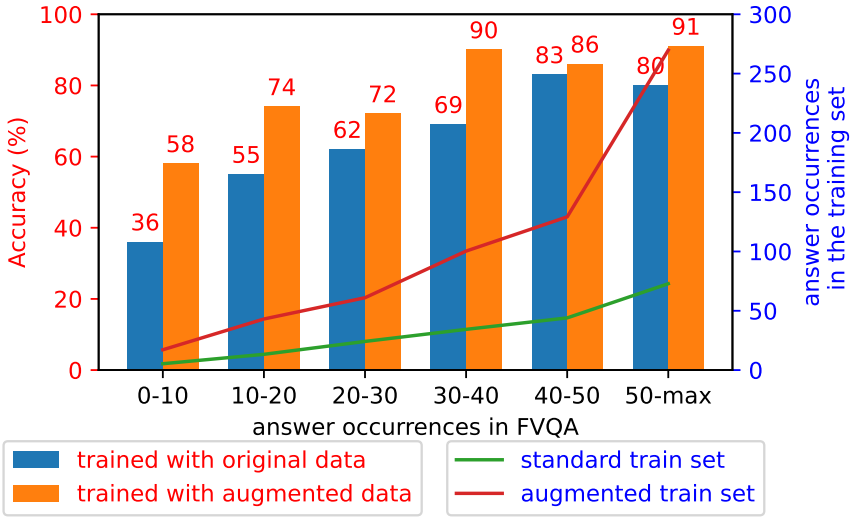


Fig. 4.4 RA-VQA-DPR accuracy on adversarial questions and answer occurrences in the standard/augmented training sets. They are grouped by the number of answer occurrences in the original FVQA dataset (binning by answer frequency). For example, a question is counted towards the '0-10' group if its answer appears less than 10 times in the original dataset.

augmentation scheme (red v.s. green curve in Fig. 4.4) compensates for the imbalanced answer distribution by providing more question variants so that systems are trained on both popular and rare answers.

4.5.6 Ablation Study

We include some additional ablation experiments in Table 4.3. It can be easily seen that the performance on originating questions (the original FVQA questions that are used to derive the adversarial samples) is very high even when images are excluded. This further supports our argument that the original dataset is heavily biased to frequent answers. The performance on the adversarial set is lower, showing that this new test set is more challenging and less biased toward language patterns.

4.6 Summary

In this chapter, we show that the FVQA test sets are not sufficiently indicative of true system performance through providing FVQA 2.0, a new human-verified adversarial test set that contains adversarial variants of the original test set questions. We show the value of

Models	Standard Test Set	Originating Question Set	Adversarial Test Set
RA-VQA-DPR	69.56 $\pm$0.78	87.52 $\pm$1.68	76.91 $\pm$1.93
(without triplets)	66.19 $\pm$1.15	84.59 $\pm$1.24	71.48 $\pm$2.08
(without images)	43.83 $\pm$0.68	57.53 $\pm$2.93	50.02 $\pm$1.00
(without triplets and images)	40.29 $\pm$1.60	51.41 $\pm$3.25	42.55 $\pm$0.90

Table 4.3 The performance of some additional baseline systems on the standard test set, originating questions (from which the adversarial questions are derived), and adversarial test set. Results are reported as the average of 5 folds with standard deviations.

adversarial samples in KB-VQA datasets by showing an augmentation scheme that leverages structural information in KGs to create augmentation questions for training, which improves models' robustness to adversarial variants.

Additionally, the work discussed in this chapter addresses research questions RQ1 and RQ2, details of which will be elaborated in the final chapter of the thesis (Sec. 9.1).

In the next chapter, we will focus on improving the ability of the retriever in the RA-VQA framework, by introducing Fine-grained Late-interaction Retrievers (FLMR).

Chapter 5

Fine-grained Late-interaction Multi-modal Retrievers

5.1 Introduction

In Chapter 3, we introduced Retrieval Augmented Visual Question Answering (RA-VQA). RA-VQA is a framework designed to answer difficult KB-VQA questions [208, 88, 183], with the most recent version achieving performance close to large models (such as GPT-3 [23]) while using much simpler models. RA-VQA first retrieves K documents relevant to the image and the question from an external knowledge base, and then generates the answer using an LLM grounded in the retrieved passages.

In this chapter, we focus on addressing two major limitations in RA-VQA's retriever:

(1) *Incomplete image understanding*: image representations are obtained via image-to-text transforms such as captioning and object detection. While effective, this approach can result in incomplete image understanding, which hinders the retrieval of relevant knowledge. This is a common issue for retrieval-based KB-VQA systems in the literature.

(2) *Lossy compression of visual scenes and questions to a single embedding*: the Dense Passage Retrieval (DPR) [143] retriever, widely used in current retrieval-based QA systems, computes relevance scores between queries and documents with their respective, one-dimensional embeddings. However, compressing complex visual scenes and questions into a single embedding can be lossy. This is especially problematic in KB-VQA, where queries and visual elements are more diverse than in other Open-domain QA tasks. DPR could overlook finer-grained relevance, resulting in degraded retrieval performance.

To address these two limitations we propose an enhanced knowledge retrieval model called Fine-grained Late-interaction Multi-modal Retriever (FLMR). FLMR incorporates

finer-grained, token-level visual and textual features into multi-dimensional embeddings. When computing relevance scores, FLMR considers the interaction between every pair of token embeddings, including cross-modality interaction between texts and images, enabling a finer-grained assessment of relevance. We also introduce large vision models such as ViT [72] to produce visual tokens that complement text-based image representations for more complete image understanding. To ensure that the interactions between visual and text tokens are well-defined, we align the vision model with the text-based retriever with a simple yet effective alignment training procedure. We also find that FLMR is able to make use of finer-grained regions of interest, leading to better recall rate, whereas DPR's recall rate degrades when these finer-grained features are incorporated. Our FLMR retriever achieves a significant increase of approximately 8% in PRRecall@5 for knowledge retrieval, and a competitive VQA score of 61%, surpassing the state-of-the-art models with the same scale of parameters.

The contributions of this chapter are outlined as follows:

- We introduce FLMR, the first-of-its-kind to leverage Late Interaction (introduced in Sec. 2.2.1) and multi-dimensional representations to capture fine-grained, cross-modality relevance that significantly improves retrieval performance over existing state-of-the-art KB-VQA retrievers;
- We show that introducing image representations from a large vision model after a simple yet effective alignment procedure can complement image representations obtained via image-to-text transforms, leading to more complete image understanding, better knowledge retrieval, and higher VQA accuracy. This offers improvements to current VQA systems as many systems have only a single mode of image understanding that relies on either image-to-text transforms or vision models;
- We achieve a substantial improvement of approximately 8% in knowledge PRRecall@5 over other state-of-the-art retrievers in the OK-VQA dataset, with an accuracy of 61% that surpasses other systems with similar parameter sizes.

The code, data, and pre-trained model weights have been released at: https://github.com/LinWeizheDragon/Retrieval-Augmented-Visual-Question-Answering.

5.2 Related Work

This section provides an overview of the related work up to May, 2023, the time of this study.

5.2.1 Visual Question Answering Systems

Recent work in VQA can be roughly divided into four categories with respect to multi-modal modeling: (1) Visual and textual features can be fused via cross-modality fusion [362, 286, 363, 133, 97]; (2) Multi-modal models can be trained from scratch to jointly understand vision and language before they are fine-tuned to perform VQA tasks [294, 45, 87, 175, 318, 370, 173]; (3) Vision model and language models that have been pre-trained on unimodal corpora can be aligned to avoid expensive multi-modal pre-training [98, 61, 287]; (4) Image-to-text transforms such as captioning can be used to transform images into texts to enable the use of text-only reasoning pipelines [183, 96, 180, 208, 341, 88, 113]. Building on these Vision-and-Language modeling techniques, our work shows that image-to-text transforms and aligned vision models can complement each other to provide more complete visual information, leading to improved performance in both knowledge retrieval and VQA.

5.2.2 Knowledge-based VQA Systems

Recent KB-VQA systems can access both structured data, such as ConceptNet and other KGs [227, 90, 168, 326, 215, 48], as well as unstructured data such as Wikipedia passages [326, 88, 96] for knowledge retrieval. LLMs can also be a source of "implicit world knowledge": KAT [96] and REVIVE [180] prompt GPT-3 to generate potential answer candidates. RA-VQA and its prior works [208, 242, 88] ground answer generation in the retrieved knowledge from external KBs to achieve excellent VQA performance. The work in this chapter improves this retriever-reader pipeline with a novel knowledge retriever which significantly improves the recall rate of knowledge retrieval as well as the final VQA performance.

5.2.3 Knowledge Retrieval

Most retrievers in QA systems are based on DPR and its variants [143, 96, 208, 96, 183, 325]. These mainly use one-dimensional embeddings and contrastive learning for training. Late Interaction models [146, 264] have recently achieved state-of-the-art performance on QA knowledge retrieval. Our FLMR extends this paradigm to work with multi-modal features and shows that incorporating finer-grained visual features, such as regions-of-interest, leads to superior retrieval performance. EnFoRe [325] retrieves a list of entities from the image, the query, and the answer candidates, and then explicitly learns scores to indicate the importance of each entity. FILIP [346] has a similar late-interaction setting but it focuses on single modal query (image-text retrieval). To the best of our knowledge, FLMR is also the first

to introduce cross-modality, token-level late interactions to compute relevance scores for multi-modal knowledge retrieval. We also propose a light-weight method that aligns a vision model with a text-based retriever to incorporate more complete multi-modal information in retrieval queries. Compared to previous approaches that rely on expensive pre-training on multi-modal datasets [36, 346], FLMR's vision-language alignment process is efficient and can be done in 4 hours with one A-100 GPU, leveraging around 1 million image-text data pairs.

5.3 Method

In this section, we introduce RA-VQA-v2, which builds upon the original RA-VQA framework (Chapter 3) but is equipped with Fine-grained Late-interaction Multi-modal Retriever (FLMR) to enhance knowledge retrieval. As illustrated in Fig. 5.1, the framework consists of two stages: Knowledge Retrieval (Sec. 5.3.1) and Answer Generation (Sec. 5.3.2).

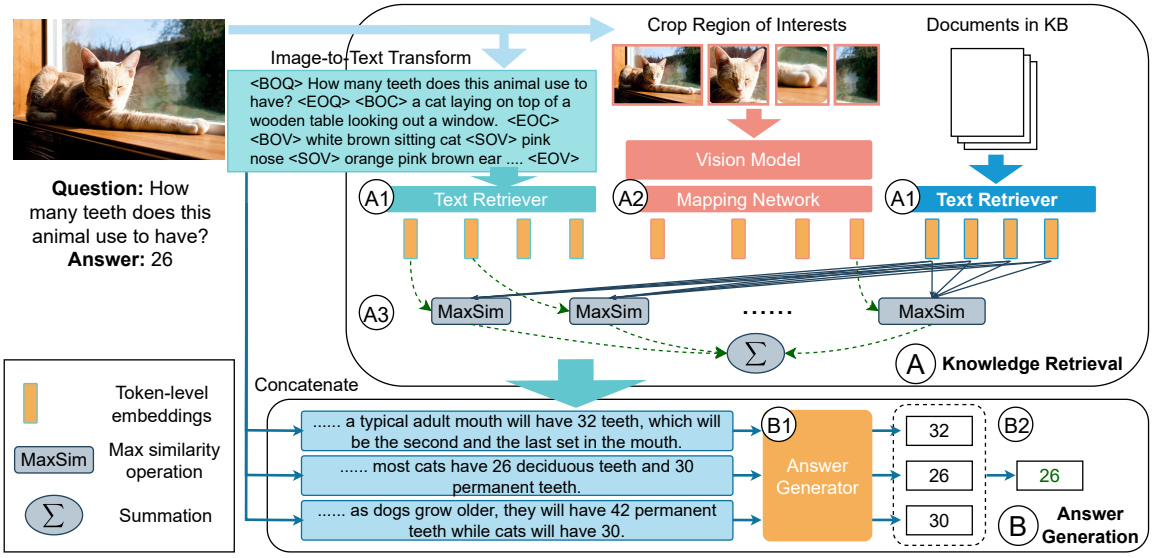


Fig. 5.1 Overview of RA-VQA-v2. The system consists of two steps: (A) Knowledge Retrieval and (B) Answer Generation. (A.1) A text retriever is used to obtain token-level embeddings of text-based vision (obtained by captioning and object detection) and text documents in the database. (A.2) Visual tokens are obtained from the image and the region-of-interest patches using a vision model and a mapping network. (A.3) Relevance score between the query and the document is computed by aggregating the fine-grained relevance at token level with late interaction mechanism (Eq. 5.3). (B.1) The answer generator takes the text query, the image, and the retrieved documents as input, generating one candidate answer per retrieved document. (B.2) The answer with the highest joint probability is selected.

5.3.1 Knowledge Retrieval

The FLMR system consists of two encoders: a vision model $\mathscr{F}_V$ and a language model $\mathscr{F}_L$ that encode image and textual features, respectively.

Visual Features

We utilise two types of visual representations: (1) text-based vision representations (textual description of visual elements) obtained by image-to-text transforms and (2) feature-based vision representations obtained by large vision models.

For text-based vision representations, to allow a direct comparison, we follow RA-VQA to extract objects and their attributes using VinVL [370] and generate image captions using Oscar [175]. For each image I, we obtain a textual description that contains serialised object names, attributes, and descriptive captions (Sec. 3.3.1). The sequence is appended to the question q to form the query. For simplicity of notation, the question q always includes text-based vision unless otherwise specified.

For feature-based vision representations, we use the vision model F_V to extract both global and regional image feature representations. For regional image feature representations, we further use the object detection results of VinVL to locate N_{ROI} (Region-of-Interest) bounding boxes. To filter bounding box proposals from VinVL, we use the predicted class name associated with each box to select objects explicitly mentioned in the question q, and then prioritise bounding boxes with larger areas. Using the vision model F_V, we then obtain one global image representation $g = \mathscr{F}_V(I) \in \mathbf{R}^{d_V}$ from the image I and ROI-based regional representations $\{r_i = \mathscr{F}_V(I_i^p) \in \mathbf{R}^{d_V}\}_{i=1,...,N_{ROI}}$ from the image ROI patches $\{I_i^p : i = 1,...,N_{ROI}\}$ which contain finer-grained details.

Token-Level Embeddings

Compared with DPR's compressed, one-dimensional representation of queries and documents, FLMR preserves richer information by employing token-level, multi-dimensional embeddings to improve retrieval.

We obtain token-level embeddings for both textual input and visual input:

For token-level text embeddings, we extract the hidden state outputs of the language model $\mathscr{F}_L(q) \in \mathbf{R}^{l_q \times d_L}$, where l_q is the length of the text query and d_L is the hidden size. As depicted in Fig. 5.1, these embeddings are 'token-level' because each of the l_q vectors (of size d_L) corresponds to a text token in the query input.

Similarly, as in Fig. 5.1, visual 'token-level' embeddings are a series of vectors that have the same dimension d_L as the text token-level embeddings but vary in length. These can be

concatenated with the text token-level embeddings to create a larger vector array. Since the visual embeddings can also be viewed as distinct vectors, we define them as 'visual tokens' to reflect their similarity to text tokens. Specifically, to align the vision and text modalities, we train a mapping network $\mathscr{F}_M$ that learns to project visual features from vision model $\mathscr{F}_V$ with hidden size d_V into the latent space of the language model $\mathscr{F}_L$ with hidden size d_L. The mapping network, a 2-layer multi-layer perceptron, projects each visual representation into N_{vt} visual tokens, i.e., $\mathbf{R}^{d_V} \rightarrow \mathbf{R}^{N_{vt}d_L/2} \rightarrow \mathbf{R}^{N_{vt}d_L}$ and finally reshaped into $\mathbf{R}^{N_{vt}\times d_L}$.

The visual token-level embeddings and text token-level embeddings are concatenated to form the final embeddings of queries and documents. Formally, the final query embeddings $\mathbf{Q}$ are:

$$\mathbf{Q} = [\mathscr{F}_L(q), \mathscr{F}_M([g, r_1, r_2, ..., r_{N_{ROI}}])] \in \mathbf{R}^{l_Q \times d_L}, \tag{5.1}$$

where $l_Q = l_q + (N_{ROI}+1) \times N_{vt}$. l_q is the length of the question q. $[v_1, ..., v_N]$ denotes the concatenation of N embeddings v_1 to v_N.

The documents in the knowledge base are represented by embeddings $\mathbf{D}$ obtained from the document content d of length l_D:

$$\mathbf{D} = \mathscr{F}_L(d) \in \mathbf{R}^{l_D \times d_L} \tag{5.2}$$

Multi-Modal Late Interaction

We compute the relevance score between a question-image pair $\bar{\mathbf{q}} = (q, I)$ and a document d by a late interaction formula similar to that in ColBERT but under the multi-modal context:

$$r(\bar{\mathbf{q}}, d) = r((q, I), d) = \sum_{i=1}^{l_Q} \max_{j=1}^{l_D} \mathbf{Q}_i \mathbf{D}_j^\top. \tag{5.3}$$

For each query token, the MAX operation selects the highest relevance score over all document tokens. In preliminary experiments, other operations (e.g. MEAN or SUM) were found to be overly sensitive to the length of documents, which can be as short as a single sentence.

In contrast to DPR, FLMR allows full interactions between every query embedding vector $\mathbf{Q}_i$ and every document embedding vector $\mathbf{D}_j$. Additionally, FLMR retriever also supports retrieving multi-modal documents. We leave the formulation and results to Appendix A.4.

Training and Inference

To train the model, similar to Chapter 3 and 4, we treat documents d^* that contain the ground-truth answer to question q as gold (positive) documents. We use in-batch negative

sampling for training following Karpukhin et al. [143]. All documents in a training batch other than d^* are considered negative for q, denoted as $\mathscr{N}(q)$. We train with the contrastive loss $\mathscr{L}_{CL}$ over the dataset $\mathscr{D}$:

$$\mathscr{L}_{CL} = -\sum_{(q,d^*)\in\mathscr{D}} \log \frac{\exp\left(r(q,d^*)\right)}{\exp\left(r(q,d^*)\right) + \sum_{z\in\mathscr{N}(q)} \exp\left(r(q,z)\right)}. \tag{5.4}$$

After training, all documents are indexed using PLAID [263], which enables fast late-interaction retrieval with a time cost similar to that of DPR.

Training the Mapping Network for Vision-Language Alignment

Directly fine-tuning the two models $\mathscr{F}_V$ and $\mathscr{F}_L$ on the retrieval task leads to performance degradation at the start of training, as the models are not yet aligned. Inspired by CLIP [246], where a language model is trained to align with a vision model, we align $\mathscr{F}_V$ and $\mathscr{F}_L$ in the context of knowledge retrieval by pre-training the parameters of the mapping network $\mathscr{F}_M$ with a retrieval task.

Given ground-truth image-document pairs $\{(I_p, d_p)\}$, which can be Wikipedia images and their accompanying texts, the system is trained to retrieve the document d_p associated with the input image I_p. The relevance between the input image I and a document d is formulated as

$$\begin{aligned} \mathbf{Q} &= \mathscr{F}_M(F_V(I)) \in \mathbf{R}^{N_{vt}\times d_L}; \\ \mathbf{D} &= \mathscr{F}_L(d) \in \mathbf{R}^{l_D\times d_L}; \\ r(I,d) &= \sum_{i=1}^{N_{vt}} \max_{j=1}^{l_D} \mathbf{Q}_i \mathbf{D}_j^\top. \end{aligned} \tag{5.5}$$

where only the parameters of the mapping network $\mathscr{F}_M$ are trained with the contrastive loss in Eq. 5.4. We provide details of pre-training in Appendix A.2 and discuss its effectiveness in Sec. 5.5.2.

Knowledge Retrieval

We extract top-K documents from the knowledge base as relevant knowledge. The retrieval probability is defined below following the notation in Sec. 3.3 and Lewis et al. [167]:

$$p_\theta(d_k|\bar{\mathbf{q}}) = \frac{\exp(r(\bar{\mathbf{q}}, d_k))}{\sum_{j=1}^{K} \exp(r(\bar{\mathbf{q}}, d_j))}, \tag{5.6}$$

where θ denotes the model parameters of $\mathscr{F}_V$, $\mathscr{F}_L$, and $\mathscr{F}_M$.

5.3.2 Answer Generation

In principle, the knowledge retrieved by FLMR can be used by any VQA answer generator. We denote the answer generator as $\mathscr{F}_A$ with parameters ϕ. Following RA-VQA, RA-VQA-v2 generates an answer for each retrieved document and selects the best candidate by the joint probability of retrieval and answer generation:

$$\{d_k\}_{k=1}^{K} = \text{topK}_d\left(p_\theta(d|\bar{\mathbf{q}})\right); \quad \widehat{y}, \widehat{d} = \underset{y,d_k}{\text{argmax}}\, p(y, d_k|\bar{\mathbf{q}}) = \underset{y,d_k}{\text{argmax}}\, p_\phi(y|\bar{\mathbf{q}}, d_k)\; p_\theta(d_k|\bar{\mathbf{q}}). \tag{5.7}$$

The training loss of the answer generator follows that of the underlying model. For example, when using BLIP 2 [171], we use the cross-entropy loss of the generated sequences:

$$\mathscr{L} = -\sum_{(\bar{\mathbf{q}},\mathscr{S})\in\mathscr{T}} \sum_{k=1}^{K} \log p_\phi(s_k^*|\bar{\mathbf{q}}, d_k), \tag{5.8}$$

where $\mathscr{T}$ is the whole dataset. $\mathscr{S}$ is the set of human responses. $s_k^* \in \mathscr{S}$ is the answer string that appears in the retrieved document d_k, or the most popular answer string[1] if an exact match cannot be found in the document.

5.4 Experiment Setup

5.4.1 Datasets

Similar to Chapter 3, we focus on the OK-VQA dataset where a large portion of questions requires external knowledge (either commonsense or domain-specific) to answer. There are no annotations of 'ground-truth' documents for OK-VQA questions. We follow the literature to use pseudo-relevance labels (a binary indicator of whether a document contains the answer string) as document annotations. We do not evaluate A-OKVQA [273], a successor of OK-VQA, as it emphasises visually-grounded reasoning rather than knowledge retrieval. To validate the effectiveness of our proposed approach, we test the retrieval abilities using 2 different corpora, whose statistics can be found in Appendix A.1:

(1) *Google Search Corpus for OK-VQA* [208]: a passage corpus collected for answering OK-VQA questions. Previous work has shown that the corpus is effective for OK-VQA [208,

[1]The most popular answer is the one chosen by most annotators.

183]. We use this corpus in evaluating VQA performance since it covers more knowledge for answering the OK-VQA questions.

(2) *Wikipedia Corpus for OK-VQA*: we collect this corpus by gathering all Wikipedia passages on common objects and concepts (e.g. umbrella, dog, hat) and those containing any of the potential answers in OK-VQA training set. This ensures the corpus covers useful knowledge for answering OK-VQA questions. We note that the collected corpus encompasses a multitude of semantically-diverse documents (>100,000) that challenge the retrieval system to identify actually useful documents. For example, all Wikipedia documents with the word 'party' are included in the corpus, ranging from descriptions of fairy tales to political parties.

We evaluate on two additional KB-VQA datasets to demonstrate FLMR's generalisability.

(1) FVQA [311]: We follow RA-VQA to preprocess the data. All knowledge triplets are serialized into text sequences to form the knowledge base for retrieval. The average of 5 cross-validation splits is reported.

(2) Infoseek [43]: Infoseek is a newly proposed KB-VQA dataset that provides Wikipedia documents that can be used in answering its questions. We follow Chen et al. [43] in pre-processing. First, we remove questions whose answers cannot be found in the provided Wikipedia passages. Second, in additional to the documents covered in the dataset ($\sim$60,000), we include less relevant passages to form a knowledge base for retrieval ($\sim$100,000 documents). The test set annotation has not been released, and so we split the official validation set again into validation and test sets ($\sim$5200 questions).

We use 10% of the WIT dataset [290], a corpus based on Wikipedia with image-text pairs, to train the mapping network for multi-modal alignment.

5.4.2 Training Setup

We use `ColBERTv2` [264] and CLIP ViT-Base [246] to initialise the text-based retriever and vision encoder. For the DPR baseline, we use the official DPR checkpoints to initialise the retriever. In answer generation, we use T5-large [248] and BLIP2-Flan-T5-XL. We use 1 Nvidia A100 (80G) for all experiments. We give detailed training hyperparameters in Appendix A.2. We use LoRA [110] to fine-tune RA-VQA-v2 (BLIP 2) on 1 single GPU. The vision model is frozen throughout all experiments. During vision-language alignment training, only the mapping network is trainable. In training the answer generator, the retriever is frozen.

5.4.3 Evaluation

We present the metrics used to assess the generated answer and the performance of our knowledge retriever below. They are the same metrics as those used in Sec. 3 and we repeat them for readers' convenience. All reported numbers are averaged from 3 runs with different seeds. We verified the significance of all mentioned improvements with `scipy.stats.ttest_ind` ($p < 0.05$).

(1) *VQA Score*: To evaluate VQA performance, we use the official VQA Score [214] which assigns score to the generated answer based on its exact occurrence count in the set of human responses $\mathscr{S}$:

$$\text{VQAScore}(y, \mathscr{S}) = \min\big(\frac{\#_{\mathscr{S}}(y)}{3}, 1\big), \tag{5.9}$$

where $\#_{\mathscr{S}}(y)$ is the occurrence of y in human responses $\mathscr{S}$. This score ensures that a model is partially rewarded even if it generates a less popular answer among the human responses [208].

(2) *Exact Match (EM)* awards point if any of the annotated answers is generated exactly: $\text{EM}(y, \mathscr{S}) = \min(\#_{\mathscr{S}}(y), 1)$.

(3) *Pseudo Relevance Recall (PRRecall@K)*: To evaluate the retriever, we adopt pseudo relevance following Luo et al. [208] due to the absence of ground-truth knowledge documents for each query. A document is considered pseudo-relevant if it contains any human-annotated answers. PRRecall@K measures whether the retrieved K documents contain at least one pseudo-relevant document: $\text{PRRecall@K} = \min\big(\sum_{k=1}^{K} H(d_k, \mathscr{S}), 1\big)$, where $H(d_k, \mathscr{S})$ evaluates to 1 if the retrieved document d_k contains any answer in $\mathscr{S}$, and 0 otherwise. The metric is averaged across the test set.

5.4.4 Baselines

To demonstrate the effectiveness of **FLMR**, we take a **DPR** retriever as a baseline. In later sections, FLMR *w/o Late Interaction* refers to the corresponding DPR baseline. We apply the same pre-training strategy, training data, and hyperparameters to construct a multi-modal retriever based on DPR. Particularly, we keep the product $N_{vt} \times d_L$ and the number of parameters of the vision mapping networks identical for FLMR and DPR for a fair comparison. Since DPR can only handle one-dimensional query and document embeddings, we sum the embeddings of the [CLS] token from $\mathscr{F}_L(\cdot)$ and the visual tokens from $F_M(F_V(\cdot))$ to reduce the dimension to $1 \times d_L$. Formally, the query and document embeddings are:

$$\begin{aligned}\mathbf{Q_{dpr}} &= \left(\mathscr{F}_{L,\text{CLS}}(q) + \mathscr{F}_M(\mathscr{F}_V(g)) + \sum_{i=1,\ldots,N_{ROI}} \mathscr{F}_M(\mathscr{F}_V(r_i))\right) \in \mathbf{R}^{d_L}, \\ \mathbf{D_{dpr}} &= \mathscr{F}_{L,CLS}(d) + \mathscr{F}_M(\mathscr{F}_V(I_d)) \in \mathbf{R}^{d_L}.\end{aligned} \tag{5.10}$$

where I_d is the image of the document if multi-modal document collection is used and otherwise omitted. The inner product search (supported by FAISS [138]) is used to train and retrieve documents with DPR.

We also compare our VQA performance with the current KB-VQA models. Amongst these models, ConceptBERT [90], KRISP [215], VRR [208], MAVEx [326], KAT-T5 [96], TRiG-Ensemble [88], and RA-VQA (Chapter 3) are relatively small in model size (<1B), whereas PICa [341], KAT [96], Prophet [277], PromptCap [113], REVIVE [180], PALI [38], Flamingo [7], PaLM-E [74] use very large models such as GPT-3 (175B) and PaLM-E (562B).

5.5 Results and Key Findings

5.5.1 VQA Performance

As shown in Table 5.1, recent models leverage LLMs or LMMs to achieve excellent performance on OK-VQA. The best performance to date is by PaLM-E, achieving a VQA score of 66.1 with 562 billion pre-training parameters. The original RA-VQA formalism achieves a lower VQA Score of 54.48 but with only 800 million parameters.

We first compare RA-VQA-v2 (with FLMR retrieval) with RA-VQA (with DPR retrieval). Compared with RA-VQA (T5-large), RA-VQA-v2 (T5-large) improves the PRRecall@5 significantly from 83.08% to 89.32%, leading to a gain of 3.4 in VQA Score (51.45 to 54.85). This suggests that improvement in knowledge retrieval benefits answer generation via retrieval augmentation.

We also show the effectiveness of knowledge augmentation by comparing the underlying base models with their retrieval-augmented version. As shown, T5-large and BLIP 2 (fine-tuned with OK-VQA data) achieve 47.52 and 55.44 VQA Scores, respectively. Their retrieval-augmented version, RA-VQA-v2 (T5-large) and RA-VQA-v2 (BLIP 2) gain 7.33 and 6.64 in VQA Score, respectively. For readers' interest, we provide more thorough analyses on the performance that the underlying answer generator model attains and the gain brought by knowledge retrieval in Appendix A.5, using the Hit Success Ratio introduced in Chapter 3.

#	Model	Base Models	K	Knowl. Src.	R@5	EM	VQA
1	ConceptBERT			C			33.66
2	KRISP			C + W			38.35
3	VRR		100	GS			45.08
4	MAVEx			W + C + GI			39.40
5	KAT-T5	T5-large	40	W			44.25
6	TRiG-Ensemble	T5-large	100	W		54.73	50.50
7	RA-VQA (joint training)	T5-large	50	GS	82.84	59.41	54.48
8	RA-VQA	T5-large	5	GS	81.25	55.77	51.22
	Systems based on large models (≥3B parameters)						
9	PICa	GPT-3		GPT-3			48.00
10	KAT-Ensemble	T5-large, GPT-3	40	W + GPT-3			54.41
11	Prophet	GPT-3		GPT-3			61.11
12	PromptCap	GPT-3		GPT-3			60.40
13	REVIVE	GPT-3		W + GPT-3			58.00
14	PALI	PALI (3B)		PALI			52.40
15	PALI	PALI (15B)		PALI			56.50
16	PALI	PALI (17B)		PALI			64.50
17	Flamingo	Flamingo (80B)		Flamingo			57.80
18	PaLM-E	PaLM-E (562B)		PaLM-E			<u>66.10</u>
	Baselines without knowledge retrieval						
19	T5-large (fine-tuned) *w/o knowledge*	T5-large				51.38	47.52
20	BLIP 2 (fine-tuned) *w/o knowledge*	BLIP 2 (T5-XL)				59.49	55.44
	Our proposed models (models w/o Late-interaction use DPR instead of FLMR)						
21	RA-VQA-v2 (T5-large)	T5-large	5	GS	89.32	58.85	54.85
22	*w/o ROI & VE & Late-interaction*	T5-large	5	GS	83.08	55.89	51.45
23	RA-VQA-v2 (BLIP 2)	BLIP 2 (T5-XL)	5	GS	**89.32**	**62.01**	**62.08**
24	*w/o ROI*	BLIP 2 (T5-XL)	5	GS	87.02	61.63	60.75
25	*w/o ROI & VE*	BLIP 2 (T5-XL)	5	GS	85.99	59.95	60.41
26	*w/o Late-interaction*	BLIP 2 (T5-XL)	5	GS	82.90	59.00	58.20
27	*w/o ROI & Late-interaction*	BLIP 2 (T5-XL)	5	GS	83.43	60.18	59.21
28	*w/o ROI & VE & Late-interaction*	BLIP 2 (T5-XL)	5	GS	83.08	59.49	58.70

Table 5.1 Model Performance on OK-VQA. Knowledge Source abbreviations: C: ConceptNet; W: Wikipidia; GS: GoogleSearch; GI: Google Images. EM stands for Exact Match. VQA stands for VQA Score. R stands for PRRecall. The best performance in literature is <u>underlined</u>.

Model	VQA Score
RA-VQA-v2 (BLIP 2)	62.03
w/o text-based vision	60.37
BLIP 2 (fine-tuned) *w/o knowledge*	55.44
w/o text-based vision	54.10

Table 5.2 Removing text-based vision from answer generation reduces the VQA performance, showing that text-based vision offers more complete image understanding.

To confirm that text-based vision can aid LMMs such as BLIP 2 which already has its own image encoder in VQA tasks, we remove text-based vision from RA-VQA-v2 (BLIP 2) and BLIP 2 (fine-tuned). This results in a decrease in VQA performance from 62.03 to 60.37 and 55.44 to 54.10, respectively (Table 5.2), suggesting that text-based vision contains useful information not included in the visual features obtained by BLIP 2's own image encoders.

RA-VQA-v2 achieves comparable and even better performance when compared with systems that use very large ($\geq$13B parameters) LLMs and LMMs. With BLIP 2 ($\approx$3B), RA-VQA-v2 outperforms Flamingo (80B) by 4.19 VQA Score. It also outperforms many recent systems that use GPT-3 (175B) as an answer generator or knowledge source, such as PromptCap, REVIVE, and KAT. It achieves similar performance to that of PALI (17B) (62.03 vs 64.5 VQA Score). With comparable parameter sizes, RA-VQA-v2 (BLIP 2, 3B) outperforms PALI (3B) by a large absolute margin (62.03 vs 52.40 VQA Score). We emphasise that RA-VQA-v2 can be used in conjunction with virtually any existing LLMs and LMMs to offer substantial improvements, as demonstrated by the T5-large and BLIP 2 experiments.

5.5.2 Retrieval Performance

Text-based v.s. Feature-based Vision

As shown in Table 5.3, previous retrievers (RA-VQA, VRR) achieve $\approx$82.84% PRRecall@5 using only text-based vision (textual descriptions of visual scenes). We show that visual features obtained via aligned vision models (feature-based vision) are equally effective as text-based vision. Relying on questions only, FLMR has a baseline retrieval score of 74.81 PRRecall@5. Incorporating text-based vision and feature-based vision increase PRRecall@5 to 85.99 and 85.08, respectively. Furthermore, feature-based vision provides information complementary to test-based vision, as demonstrated by the better PRRecall@5 at 87.02 when the two are combined. The same trend is observed for the DPR-based retrieval system, though less dramatically (from 83.08 to 83.43). We note that pre-training the mapping

#	Retriever	Text-based Vision	Feature-based Vision	GS		Wikipedia	
				R@5	R@10	R@5	R@10
1	VRR	✓	-	80.4	88.55		
2	RA-VQA-FrDPR	✓	-	81.25	88.51		
3	RA-VQA	✓	-	82.84	89.00		
4	DPR	-	-	73.11	82.05	57.03	69.84
5	DPR	✓	-	83.08	89.77	66.04	75.94
6	DPR	-	✓	80.52	88.27	65.84	75.85
7	DPR	✓	✓	83.43	90.31	66.88	76.35
8	DPR	✓	✓+9ROIs	82.90	89.95	65.86	75.90
9	FLMR	-	-	74.81	83.10	57.20	70.11
10	FLMR	✓	-	85.99	92.79	66.50	76.80
11	FLMR	-	✓	85.08	91.80	66.90	77.05
12	FLMR	✓	✓	87.02	92.69	67.50	77.60
13	FLMR	✓	✓+9ROIs	**89.32**	**94.02**	**68.10**	**78.01**
14	*w/o alignment pre-training*	✓	✓+9ROIs	85.71	92.41	66.40	76.10

Table 5.3 Retrieval performance on Google Search (GS) and Wikipedia. Text-based vision refers to textual descriptions of images (such as OCR, caption, objects and attributes). Feature-based vision is obtained using a neural vision model directly (e.g. ViT). R@K refers to PRRecall@K.

network for vision-language alignment is crucial for good performance. Without such pre-training, performance degrades to 85.71. These results confirm that incorporating aligned vision encoders in the retrieval process compensates for the information loss in image-to-text transforms.

Effects of Late Interaction and ROIs

Late Interaction enables FLMR to capture fine-grained relevance of token-level embeddings. As shown in Table 5.3, using the same query and document representations, upgrading DPR to FLMR leads to consistent improvement in retrieval performance by a large margin up to $\sim$6% (comparing Table 5.3 Row 8 & 13).

In addition to token-level relevance, FLMR can utilise fine-grained Region-of-Interest (ROI) features with Late Interaction whereas DPR can not. This can be demonstrated by Fig. 5.2: as the number of ROIs increases, DPR performance degrades. This may be because DPR's one-dimensional query and document embeddings are not expressive enough to encompass fine-grained details of the ROI visual cues. As shown in Table 5.3 and Table 5.1 Row 27-28, adding more ROIs effectively adds noise which adversely impacts the retrieval performance (83.43 to 82.9), and in turn worsen VQA scores (59.2 to 58.2).

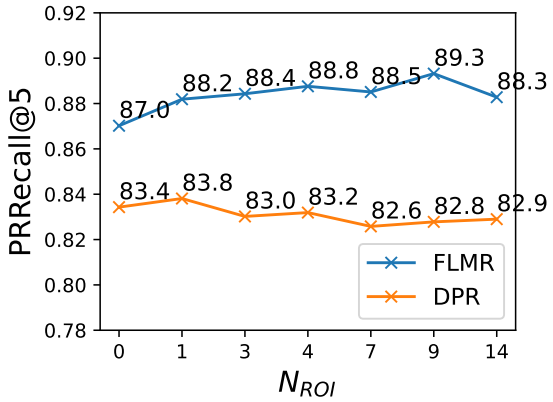


Fig. 5.2 PRRecall@5 versus the number of ROIs. Finer-grained ROIs cause performance degradation in DPR, while FLMR captures them to improve retrieval performance.

Object-centric ROIs improve retrieval

	PRecall@5	PRecall@10
4 Object-centric ROIs	88.01	93.62
4 Random ROIs	86.9	93.20
4 Evenly-split ROIs	86.96	93.16

Table 5.4 Comparison of ROI selection methods.

We also conduct ablation studies to show that the performance improvements brought by ROIs come from the finer-grained information captured by them rather than from increases in the number of features. We compare FLMR with 4 object-centric ROIs (obtained by object detection) against 2 baseline ROI selection methods: (1) randomly crop 4 patches of size larger than 100×100 from the image as ROIs; (2) evenly split the image to obtain the top-left, top-right, bottom-left, and bottom-right of the image as ROIs. As shown in Table 5.4, FLMR with 4 ROIs from VinVL object detection outperforms others, suggesting that it is the object-centric, fine-grained ROIs that improve the performance.

Retrieval performance on FVQA and Infoseek

As shown in Table 5.5, we observed similar improvements with FLMR. FLMR with both text- and feature-based vision improves DPR by 2.3% and 1.54% PRecall@5 on FVQA and Infoseek, respectively. Incorporating ROI features further improves its performance to 72.37 on FVQA and 47.08 on Infoseek. This suggests that FLMR is generalisable to other KB-VQA retrieval tasks and can bring steady improvements relative to baselines.

	FVQA Recall@5(Std.)	Infoseek PRecall@5
DPR	68.58(0.01)	44.88
FLMR (Visual Encoder)	70.88(0.01)	46.42
FLMR (Visual Encoder & 9 ROIs)	**72.37**(0.01)	**47.08**

Table 5.5 Retrieval performance on FVQA [311] and Infoseek [43]. Average recall on 5 splits is reported for FVQA. FLMR outperforms DPR trained with the same data with a clear margin.

Qualitative analysis of FLMR retrieval

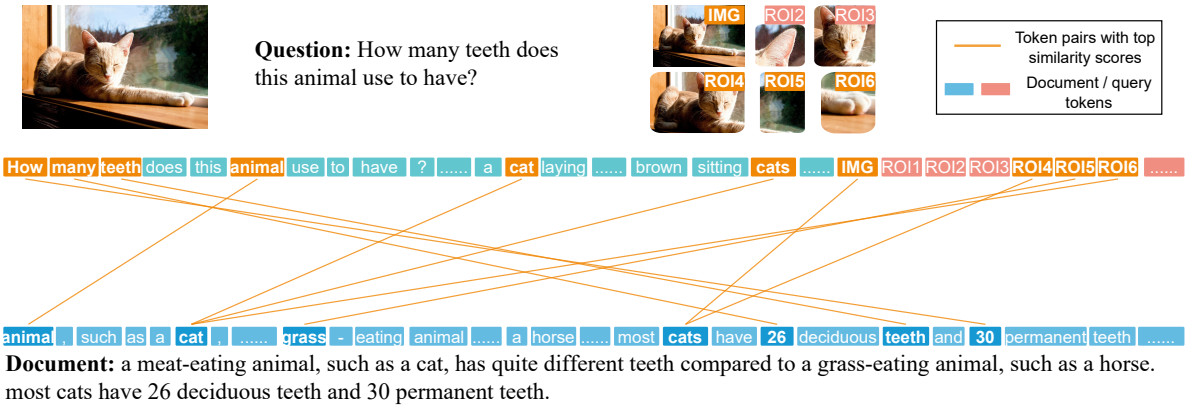


Fig. 5.3 Selected query tokens connected by document tokens that have the highest token-level relevance with them, as computed by FLMR. For example, amongst all document tokens, '26' and '30' have the highest relevance with the query token 'how' and 'many', respectively. This shows that FLMR can capture fine-grained document relevance. Zoom in for better visualisation.

Figure 5.3 shows FLMR retrieval in action. The orange lines connect the query token and the document token with the highest relevance score, which will be preserved after the MaxSim operation and will contribute to the final retrieval score. We can see that token-level interaction indeed captures fine-grained relevance between the query and the document. For example, the retriever recognises that the numbers "26" and "30" in the document are highly relevant to "How many" in the query. We can also see that the image tokens are aligned with the text tokens: the image tokens corresponding to the cat (IMG, ROI3 and ROI4) point to the words "cats" and "cat" in the document. This demonstrates the effectiveness of vision-language alignment that gives rise to explainable cross-modality relevance.

Example system outputs are presented in Fig. 5.4. It compares the model outputs and provides explanations for each case.

5.6 Limitations and Potential Future Work

The incorporation of Late Interaction retrieval introduces additional latency for both training and inference. The computational cost is evaluated in Appendix A.6. We note that in practice, the feature extraction process of multiple ROI features is computationally expensive. In Chapter 6, we will investigate alternative model designs that maintain high performance levels while obviating the need for ROI features.

5.7 Summary

In this chapter, we proposed Fine-grained Late-interaction Multi-modal Retrieval (FLMR), the first of its kind to leverage fine-grained token-level relevance between queries and documents for VQA tasks. FLMR incorporates feature-based vision using an aligned vision model that complements text-based vision to enhance image understanding, improve retrieval performance and advance VQA performance. We achieve superior performance in OK-VQA, greatly surpassing previous systems with similar parameter size and closes the gap with those systems utilizing very large ($\geq$13B) models.

Additionally, the work discussed in this chapter addresses research questions RQ1 and RQ2, details of which will be elaborated in the final chapter of the thesis (Sec. 9.1).

In the next chapter, we will investigate the scaling behavior of FLMR from multiple perspectives. As a result, we will introduce a general-purpose multi-modal late-interaction retriever, PreFLMR (Pre-trained FLMR), that achieves substantial performance improvement relative to the original FLMR.

Question: On what holiday do people traditionally eat this bird?
Caption: a large chicken walking through a field of grass.

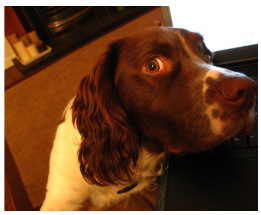
RA-VQA

	Chosen Prediction
christmas dinner? these mouthwatering recipes ... the sweet and spicy turkey glaze gives the bird a wonderful flavor. it's been a tradition in my home for as long as i can remember.	Christmas ✗
turkey doesn't have to rule the roost at your holiday feast.this extremely french way of cooking duck is not recommended for thanksgiving.	Thanksgiving
on thanksgiving, most of us will sit down to feast on a turkey dinner. the bird also shows up on the table at christmas. how did we end up with the tradition of eating turkeys during the holidays?	Thanksgiving

RA-VQA-v2

	Chosen Prediction
the same traditional thanksgiving menu, with turkey, cranberry sauce, stuffing, and pumpkin pie taking up the most real estate on our plates.	Thanksgiving ✓
on thanksgiving, most of us will sit down to feast on a turkey dinner. the bird also shows up on the table at christmas. how did we end up with the tradition of eating turkeys during the holidays?	Thanksgiving
after 1863, the year when president lincoln made thanksgiving day a national holiday, turkeys began to land on dinner plates across the country.	Thanksgiving

Explanation: RA-VQA-v2 succcecssfully captures "turkey" and all retrieved documents are about "turkey" and "tradition". In contrast, RA-VQA, without fine-grained information, gives high scores to noisy documents.



Question: What sense is this animal known for?
Caption: a brown and white dog sitting in front of a laptop computer.

RA-VQA

	Chosen Prediction
the sense organs are the body organs by which humans are able to see, smell, hear, taste, and touch or feel.	see
researchers have now discovered that humans have a much better memory to recognize and remember sequential information.	memory ✗
over the years researchers have found a general understanding of the extent to which animals possess things such as: • language/communication • memory • cognition • emotions • intelligence and in attempting to answer the above question the jury is, well, still out.	smell

RA-VQA-v2

	Chosen Prediction
pixabay of the five senses, smell is a dog's predominant sense.	smell ✓
hound dog. known for their powerful sense of smell and great speed, hounds were historically used by hunters to track and chase prey.	smell
after 1863, the year when president lincoln made thanksgiving day a national holiday, turkeys began to land on dinner plates across the country.	smell

Explanation: RA-VQA fails to capture all three key concepts “dog”, “animal” and “sense” in the query, which results in retrieving noisy, less relevant documents.

Question: What animal is the woman's shirt replicating the pelt?
Caption: a group of people sitting at a table in a restaurant.

RA-VQA-v2 w/o Region of Interest

	Chosen Prediction
nature normally favours the males of the animal kingdom in the looks department, as a tool for attracting a mate.	bear
once the bird had been plucked, it was obvious that the right half of the skeleton was much bigger than the left.	bird ✗
if you're ready to give industries that abuse animals the bird, here are some items to avoid and suggestions about what to purchase instead: boas and feather dustersan eyewitness investigation of the largest ostrich slaughter companies in the world showed that ...	ostrich

RA-VQA-v2

	Chosen Prediction
later, animal prints mimicking the leopard, zebra, and cheetah would soon become items of fashion in the western world. printify's t-shirt dresses are tagless, custom cut and sewn to match every style.	leopard ✓
ones which replicate the fur or skin of animals like leopards, tigers, zebras, giraffes, hyenas, monkeys and much more. after the animal it imitates, for instance leopard print skirts, zebra print top and so on.	leopard
animal print is a clothing and fashion style in which the garment is made to resemble the pattern of the skin and fur of an animal such as a leopard, cheetah, snow leopard, jaguar, zebra, tiger, clouded leopard, margay, ocelot, spotted hyena, striped hyena, african wild dog, constrictor snake, giraffe or monkey.	leopard

Explanation: without fine-grained region-of-interest-based features, FLMR fails to locate the "woman" and "shirt" in the image. When ROI features are added, FLMR successfully matches "leopard" from the image representations.

Retrieved Knowledge
Correct Final Prediction
Wrong Final Prediction

Fig. 5.4 Example system outputs comparing some model variants. Explanations are given to each case. Please zoom in for the best visualisation.

Chapter 6

Scaling Up Fine-Grained Late-Interaction Multi-modal Retrievers

6.1 Introduction

In the last chapter, we introduced FLMR, which uses multi-dimensional embedding matrices to represent documents and queries and then efficiently computes their relevance scores via late-interaction (Eq. 5.3) [146], thus capturing fine-grained relevance at the token level rather than at the passage level, as in Dense Passage Retrieval (DPR) [143]. As a late-interaction retriever, FLMR substantially outperforms DPR on a range of KB-VQA tasks, with only minor speed penalties.

In all of these methods, model and data size are important considerations. There has been much work in scaling up Large Language Models (LLMs) [141, 5, 39], showing the benefits of scaling. In retrieval, Ni et al. [230] observes improvements in text retrieval models that use one-dimensional embeddings by increasing the size of text encoders. However, the scaling properties of multi-modal retrieval systems, especially under the late-interaction setting, have not been studied. In this chapter, we aim to fill this gap by investigating the following three aspects of FLMR:

(1) *Vision & Text Encoding*: The original FLMR model uses a ViT-Base model as the visual encoder that has less than 100 million parameters. We investigate how its retrieval performance is affected by scaling the size and complexity of vision and text encoders.

(2) *Pre-training*: As originally formulated (Sec. 5.3.1), FLMR employs simple, lightly trained Multi-Layer Perceptrons (MLP). We investigate whether gains can be had through more extensive model pre-training.

(3) *Task Diversity*: Though FLMR was proposed to handle retrieval in OK-VQA [273], the late-interaction formulation is readily extensible to tasks beyond KB-VQA. Rather than training on a single task/dataset, we investigate whether a general-purpose multi-modal late-interaction retriever can be created by training the model on a larger collection of tasks, including Image-to-Text, Question-to-Text, and Image&Question-to-Text retrieval tasks.

To enable our research, we gather nine open-source vision-language datasets into a suite of benchmark tasks, M2KR, for assessing Multi-task Multi-modal Knowledge Retrieval. M2KR encompasses Image-to-Text, Question-to-Text, and Image&Question-to-Text retrieval tasks, and also includes prompting instructions that can be provided to an LLM for each of the component tasks.

In this chapter, we will use M2KR to train a series of FLMR-based multi-task multi-modal retrievers. We refer to these models as PreFLMR (for Pre-trained FLMR). PreFLMR models are created by training on the entirety of the M2KR training data and these models can then be evaluated on any or all of the included tasks. PreFLMR can be used directly in its pre-trained form for multi-task multi-modal retrieval. PreFLMR can also be fine-tuned for specific M2KR tasks using the task-specific tuning data included in the collection. In both uses we find that PreFLMR gives us substantial gains across the M2KR tasks.

The focuses of this chapter are:

- The M2KR task suite encompassing nine datasets and three types of retrieval tasks for training and evaluating general-purpose vision-language retrievers. We create M2KR by re-purposing various vision and language data sets that might not be originally created for KB-VQA, thus ensuring a rich and diverse collection.
- PreFLMR, a strong multi-modal retriever pre-trained on a vision-language corpus of over ten million items. We show that PreFLMR performs well across a range of knowledge retrieval tasks when given the appropriate instructions.
- A study of the scaling behaviour of FLMR in terms of its model parameters and training data. To our knowledge, this is the first systematic study of scaling in late-interaction based vision-language retrievers and should provide empirical guidance for future work.

The code, data, and pre-trained model weights have been released at: https://preflmr.github.io/.

6.2 Related Work

This section provides an overview of the related work up to February, 2024, the time of this study.

6.2.1 Document Retrieval

As introduced in Chapter 2.2.1, DPR has become a cornerstone in knowledge-intensive tasks [30, 124, 99, 161, 167] as well as in KB-VQA tasks due to its fast and precise retrieval capabilities [143, 96, 208, 183, 325]. Recent developments in retrieval methods, particularly Late Interaction models [146, 264], have shown notable performance gains over DPR, albeit with some efficiency trade-offs [187, 188].

In multi-modal retrieval, FILIP [346] (Sec. 2.2.2, Fig. 2.27) used pre-trained late interaction models for single-modal image-text retrieval, while our FLMR system (Chapter 5) extended the approach to multi-modal retrieval for KB-VQA with finer-grained visual and text features. This chapter further extends FLMR and explores its scaling properties in multi-modal retrieval.

Another line of relevant research is KB-VQA retrieval involving Named Entities, where retrieved documents must identify the person in the image. As discussed in Sec. 2.2.3, Lerner et al. [164] trains the retriever with a multi-modal inverse cloze task, while Lerner et al. [165] shows that combining mono- and cross-modal retrieval improves performance. Both use single-dimensional embeddings to represent multi-modal queries and documents. In contrast, our work trains a single multi-modal late-interaction retriever, allowing rich token-level information interactions in retrieval.

Similar to our M2KR benchmark, A concurrent work [319] introduces M-Beir, which combines several retrieval tasks and can also be used to train and evaluate universal multi-modal retrievers. In contrast, our M2KR benchmark emphasises document retrieval in knowledge-intensive tasks and we use it to develop new models.

6.2.2 Knowledge-based VQA Systems

In Sec. 2.4.4, we discussed recent multi-modal systems that have significantly improved in complex KB-VQA tasks like OK-VQA that require external knowledge sources [227, 90, 168, 326, 215, 48, 88, 96, 116, 251], while another line of work use LMMs to directly answer knowledge-intensive questions with their internal knowledge. However, challenges remain in answering more knowledge-intensive questions [42, 218], underscoring the need for robust document retrieval. Mensink et al. [218] showed that even state-of-the-art LLMs perform poorly on difficult KB-VQA questions, with an accuracy of under 20% when retrieval is not incorporated. Our work RA-VQA (Chapter 3), RA-VQA-v2 (Chapter 5), and prior work [183, 208, 242, 88, 116, 218], demonstrated strong performance in KB-VQA by using external knowledge databases. This motivates our work to build stronger retrievers for KB-VQA to make LMMs more capable of handling knowledge-intensive tasks.

6.2.3 Scaling Retrieval Systems

Previous work has explored scaling laws in language/vision systems [141, 5], revealing correlations between model performance, computation, number of parameters, and dataset sizes. They all suggest that more parameters, more training data, and longer training time often lead to improved performance. In retrieval, Ni et al. [230] and Hu et al. [116] both observe improvements in models with one-dimensional embeddings by increasing the size of language/vision encoders. Ni et al. [230] report a significant improvement when migrating from a T5 [248]-Base text encoder to larger ones (T5-large, T5-XL, T5-XXL). Hu et al. [116] scale up from T5-Base and ViT-B(ase) to T5-Large and ViT-g [50] to improve the performance substantially. Our work reports similar scaling investigations in multi-modal late-interaction retrieval.

6.3 The M2KR Benchmark Suite

In this section, to properly study general-purpose multi-modal retrievers, we introduce the Multi-task Multi-modal Knowledge Retrieval (M2KR) benchmark suite, which will be used to train and evaluate our proposed PreFLMR model.

We convert nine diverse datasets, originally designed for vision and language tasks such as image recognition, image captioning, and conversational interactions, into a uniform retrieval format. Details of the pre-processing steps, data partition, and prompting instructions are provided in Appendix B.1.

6.3.1 Tasks and Datasets

Table 6.2 shows the composition of M2KR. We pre-process the datasets into a uniform format and write several task-specific prompting instructions for each dataset. The M2KR benchmark contains three types of tasks:

(1) Image to Text (I2T) retrieval. These tasks evaluate the ability of a retriever to find relevant documents associated with an input image. Component tasks are WIT [290], IGLUE(-en) [24], KVQA [275], and CC3M [278]. CC3M is included in the M2KR training set to improve scene understanding but not in the validation/test set as the task concerns caption generation, not retrieval. The IGLUE test set, which is a subset of WIT and has an established benchmark for assessing I2T models, is included to enable comparison with the literature. The KVQA task, initially designed as a KB-VQA task, has been re-purposed into an I2T task for our modelling purposes (Appendix B.1.1).

(2) Question to Text (Q2T) retrieval. This task is based on MSMARCO [16] and is included to assess whether multi-modal retrievers retain their ability in text-only retrieval after any retraining for images.

(3) Image & Question to Text (IQ2T) retrieval. This is the most challenging task which requires joint understanding of questions and images for accurate retrieval. It consists of these subtasks: OVEN [112], LLaVA [198], OK-VQA [273], Infoseek [44], and E-VQA [219]. We note in particular that we convert LLaVA, a multi-modal conversation dataset, into a multi-modal retrieval task (Appendix B.1.3).

Table 6.1 provides examples from each dataset, demonstrating the transformation from their original to the adapted structure.

The training/validation/test examples are downsampled from the respective sets of the original datasets. We take test examples from the original validation sets for LLaVA and Infoseek since LLaVA has no test sets and the test set annotation of Infoseek has not been released. We limit the maximum test samples to 5,120 for each dataset to allow faster performance tests on all 9 datasets. Data preprocessing and partitioning details are in Appendix B.1. We further verified that there are no identical images between the training and test sets by checking the MD5 of the images, thereby preventing data contamination during training. We use the validation splits to select hyperparameters for the models, which can be found in detail in Appendix B.2.2.

6.3.2 Evaluation

We use *Recall@K (R@K)*, which measures whether at least one of the target documents is in the top-*K* retrieved entries, to evaluate retrieval performance. For the datasets Infoseek, E-VQA, and OK-VQA, we follow previous chapters (Chapter 3-5) to employ *Pseudo Recall/PRecall@K (PR@K)* (Eq. 3.11) for evaluation. This metric measures whether at least one of the top K documents includes the target answer.[1]

We use R@10 for WIT and MSMARCO, and R@1 for LLaVA and IGLUE. Other datasets are evaluated with R@5 or PR@5. As in Table 6.3, we also report the average rank (*A.R.*) of each model over all datasets to indicate multi-task retrieval performance relative to other models in comparison; lower is better.

[1] In practice, PRecall@K more accurately reflects actual retrieval performance and exhibits a stronger correlation with the ultimate VQA performance. This is because document annotations are frequently incomplete, and alternative documents within the corpus can often provide answers to the questions.

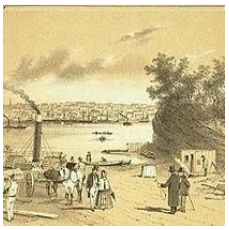
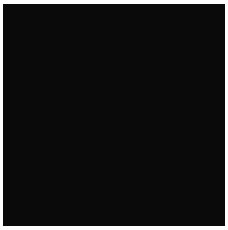
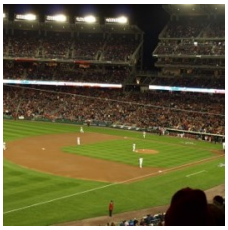
WIT	IGLUE	KVQA	CC3M	MSMARCO
				
Describe the image concisely.	Summarise the visual content of the image.	Provide a brief description of the image and the relevant details of the person in the image.	Describe the image concisely.	Retrieve the document that answers this question: how many years did william bradford serve as governor of plymouth colony?
title: PS Herald section title: Formation and operation of the North Shore Steam Company ...	title: National Library of Uzbekistan hierarchical section title: National Library of Uzbekistan caption ...	This is an image of Pilkington playing for Cardiff City in 2016. Anthony Pilkington date of birth is ...	olive oil is a healthy ingredient used liberally.	William Bradford (c.1590 - 1657) was an English Separatist leader in Leiden, ...
LLaVA	OVEN	E-VQA	Infoseek	OKVQA
				
Provide a brief description of the image along with the following question: what unique situation is occurring in this soccer match?	Using the provided image, obtain documents that address the subsequent question: what is this park called?	Obtain documents that correspond to the inquiry alongside the provided image: how big can this plant become?	With the provided image, gather documents that offer a solution to the question: What is the country of origin of this food?	Using the provided image, obtain documents that address the subsequent question: How many teeth does this animal use to have?
In this soccer match, a unique situation is occurring where three men are playing against each other, each wearing a different colored uniform.	Nationals Park is a baseball stadium along the Anacostia River in the Navy Yard neighborhood...	Dwarf cornel is a rhizomatous herbaceous perennial growing to 20cm (8 inches) tall...	title: Submarine sandwich content: Submarine sandwich A submarine sandwich, also known as a sub...	Most cats have 26 deciduous teeth and 30 permanent teeth.

Table 6.1 Demonstration of the retrieval tasks for each dataset. We show the image (first row) query, the text query (second row), and the retrieved ground truth document (third row) for each dataset. Since some retrieved documents are long, we only show part of the document and use ... to stand for continuing documents. We sampled one instruction for each dataset for demonstration. Refer to Appendix B.1 for the full list of instructions.

	#Examples			#Passages	
Datasets	Train	Val	Test	Train	Val/Test
I2T Retrieval					
WIT	2.8M	20,102	5,120	4.1M	40K
IGLUE	-	-	685	-	1K
KVQA	65K	13,365	5,120	16.3K	4,648
CC3M	595K	-	-	595K	-
Q2T Retrieval					
MSMARCO	400K	6,980	5,120	8.8M	200K
IQ2T Retrieval					
OVEN	339K	20,000	5,120	10K	3,192
LLaVA	351K	-	5,120	351K	6,006
OK-VQA	9K	5,046	5,046	110K	110K
Infoseek	676K	-	4,708	100K	100K
E-VQA	212K	9,852	3,750	50K	50K

Table 6.2 Datasets in M2KR Benchmark Suite.

6.3.3 Baselines and Systems for Comparison

For each dataset, we show the best published results in recent literature as points for comparison, if available (Table 6.3). For datasets without previous results such as LLaVA and OVEN, we use our replication of CLIP [246] and FLMR as baselines following Lin et al. [180].

6.4 PreFLMR Architecture and Training

PreFLMR's architecture is shown in Fig. 6.1. It generally follows the formulation of FLMR (Sec. 5.3.1).

PreFLMR also uses token embedding matrices $\mathbf{Q}$ and $\mathbf{D}$ to represent query and document, respectively. Given a query $\bar{\mathbf{q}}$ consisting of texts q and an image I, PreFLMR uses a language model $\mathscr{F}_L$ to obtain embeddings of all tokens in q, a vision model $\mathscr{F}_V$ to obtain embeddings of I, and a mapping structure $\mathscr{F}_M$ to project image embeddings into the text embedding space. All token-level embeddings are concatenated to form the query representation $\mathbf{Q}$. The document matrix $\mathbf{D}$ is obtained similarly with the language model $\mathscr{F}_L$ but without visual features.

The relevance score $r(\bar{\mathbf{q}}, d)$ is computed via *late-interaction* [146] between $\mathbf{Q}$ and $\mathbf{D}$, aggregating the maximum dot products over all query tokens with respect to all document

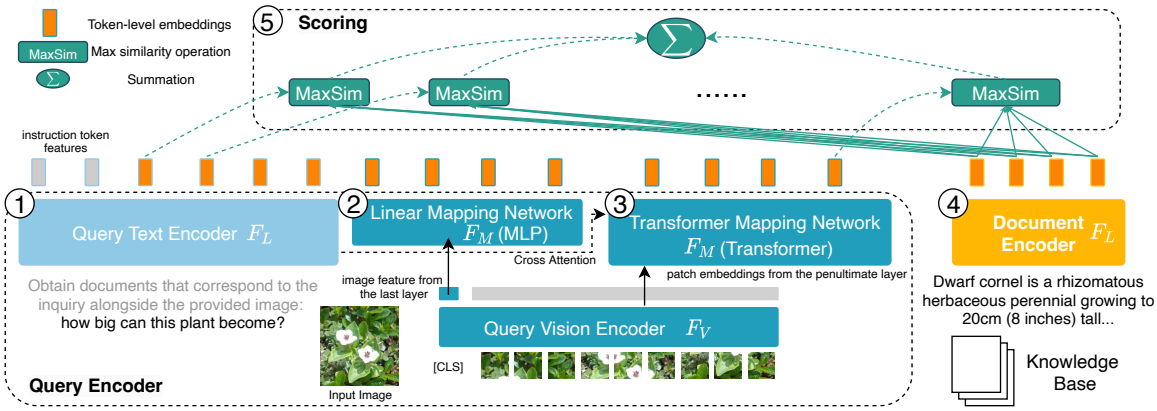


Fig. 6.1 PreFLMR Model Architecture. The grey rectangle above the Query Vision Encoder indicates the unused last layer patch embeddings. These are not utilised because only the first '[CLS]' token in the last layer of the frozen pre-trained Vision Encoder received pre-training. (1) the text query consists of an instruction and a question, which is encoded by a text encoder; (2) at the output of the vision encoder, a mapping network consisting of Multi-Layer Perceptrons (MLP) converts the '[CLS]' token representations into the same embedding space as the text encoder; (3) the transformer blocks take in the patch image embeddings from the penultimate layer of the vision encoder and attend to the text features by cross-attention; (4) a text encoder encodes documents in the knowledge base; (5) the scores between queries and documents are computed based on late-interaction, allowing each query token to interact with all document token embeddings.

tokens (Eq. 5.3). l_Q and l_D denote the total number of tokens in query $\bar{\mathbf{q}}$ and document d, respectively:

$$r(\bar{\mathbf{q}}, d) = \sum_{i=1}^{l_Q} \max_{j=1}^{l_D} \mathbf{Q}_i \mathbf{D}_j^\top. \tag{6.1}$$

PreFLMR's formulation differs from FLMR in the following aspects:

(1) While FLMR only uses the '[CLS]' embedding from ViT as the image representation, in PreFLMR we additionally extract embeddings of image patches from ViT's penultimate layer to obtain a detailed visual representation. FLMR extracts N_{ROI} Region-of-Interest (ROI) features (Sec. 5.3.1). ROI features are no longer used due to the incorporation of image patch embeddings. Another reason for disabling the use of ROI features is that it calls the visual encoder N_{ROI} times, introducing significant computational overheads in practice.

(2) We introduce Transformer blocks with cross-attention into the mapping structure to obtain query-aware visual representation. The Transformer blocks take the image patch embeddings as input, and use cross-attention to integrate the features of the text encoder. This allows PreFLMR to attend to different aspects of the image under different queries. These Transformer blocks are placed in parallel with FLMR's 2-layer MLP mapping structure.

(3) We append task-specific instructions to the text query to distinguish between tasks. The list of instructions for each task can be found in Appendix B.1. For each query, the instruction is randomly sampled from the corresponding instruction list. Instruction tokens are masked in computing relevance score. For Q2T retrieval training, we feed a blank image as PreFLMR's image input. For I2T retrieval training, we use instructions as text input to PreFLMR.

PreFLMR training and inference follow that of FLMR (Sec. 5.3.1). When training on data consisting of several datasets, we randomly shuffle the entire training data and only use in-batch negative examples from the same corpus (see Sec. 5.3.1 for the details of in-batch negatives).

6.4.1 Training Procedures

PreFLMR's pre-training involves four stages. A detailed breakdown of the data used in each stage can be found in Appendix B.2.1.

Stage 0: Text Encoder Pre-training. We train ColBERT following Khattab and Zaharia [146] on the MSMARCO dataset to obtain the initial checkpoint for PreFLMR's text encoder $\mathscr{F}_L$. This is a straightforward replication of ColBERT used as an initial text encoder as was done in FLMR, but also allowing for model size variations.

Stage 1: Training the Mapping Structure. In this stage, we only train the mapping structure $\mathscr{F}_M$, keeping the language and vision models frozen. This approach is an extension of the FLMR methodology, incorporating a larger dataset and an additional cross-attention mapping layer. The training is performed on the IQ2T dataset (LLaVA, OVEN), I2T datasets (WIT, CC3M, KVQA), and Q2T dataset (MSMARCO). Our objective is to encompass all three task types in M2KR without the need to optimise the data mixing ratio or manually select datasets to achieve an effective mapping structure. This strategy is inspired by previous studies [388, 198], which utilised relatively simple multi-modal tasks to develop image-to-text mappings.

We mask the late-interaction token embeddings in query matrix $\mathbf{Q}$ that are produced by the language model (not the token embeddings at the input embedding layer). This encourages the Transformer cross-attention layer to integrate information from its textual inputs and enables PreFLMR to perform IQ2T, I2T, and Q2T retrieval when provided with the appropriate instructions for each task.

Stage 2: Intermediate KB-VQA Pre-training. We tune the text encoder $\mathscr{F}_L$ and the mapping structure $\mathscr{F}_M$ on the E-VQA dataset, a large and high quality KB-VQA dataset, to enhance PreFLMR's retrieval performance. Including an intermediate pre-training stage to align the model with in-domain data has been well-explored in the literature (e.g., Google's TAPAS [77]). We opt for a straightforward procedure to train on E-VQA in the intermediate stage because of its diversity, increased difficulty, and larger quantity compared to other KB-VQA datasets. Specifically, E-VQA requires recognition of less common entities such as spotted hyenas and relies on more specialised domain knowledge such as American landmarks, making it good for retrieval training. This design choice is well-supported by experimental results (Table 6.3 #8 vs #5, #3 vs #2) and we provide detailed analysis in Sec. 6.5.6.

Stage 3: Full-scale Fine-tuning. We train on the entire M2KR corpora, including OK-VQA and Infoseek. This stage is straightforward multi-task learning. We tune the entire model except the vision encoder $\mathscr{F}_V$. We adjust the dataset proportions to ensure balanced learning on these datasets of varying sizes (Appendix B.2.1). Additionally, we use separate text encoders to encode queries and documents; their parameters were shared in previous steps.

6.4.2 Training Configurations

We use the Adam optimizer [149] with a fixed learning rate of 10^{-4} for the mapping structures and 10^{-5} for other parameters in all experiments. Training was run up to 300k, 220k, 12k, and 50k steps in the four stages, respectively. Full training configurations can be found in Appendix B.2.2.

6.5 Experiments and Results

In this section we present results of scaling PreFLMR components (Sec. 6.5.2, 6.5.4), analyse the effect of each training stage (Sec. 6.5.3, 6.5.6), and evaluate on the downstream KB-VQA tasks (Sec. 6.5.5). We summarise our findings in Sec. 6.5.7. Multi-task performance refers to the performance of PreFLMR models without any single-task fine-tuning.

6.5.1 Model Variants

We experiment with a range of model configurations. Model sizes range from BERT-Small (28.8M), BERT-Medium (41.1M), BERT-Base (110M) to BERT-large (340M). ColBERT text encoders are denoted as "[BERT size]-[pre-training scheme]". There are two ColBERT pre-training schemes: "v1" [146] and "v2" [264]. "v2" yields a better performing model than "v1" as evaluated on MSMARCO. We compare models initialised from "v1" and "v2" checkpoints to investigate how the performance of the initial uni-modal text retriever affects the final multi-modal vision-language retriever. Except for "Base-v2", all ColBERT variants are trained using our replication of ColBERT following the "v1" pre-training scheme.[2] For the vision encoders, we use the ViT variants: ViT-B(ase) (88M) [246], ViT-L(arge) (303M) [246], ViT-H (631M) [50] and ViT-G (1.84B) [50].

6.5.2 PreFLMR Performance

The best-performing PreFLMR model (ViT-G + Base-v2) outperforms other variants on most of M2KR benchmark (Table 6.3, #13). Without single-task fine-tuning, PreFLMR outperforms baseline models optimised for the individual tasks on 7 out of 9 datasets, showcasing its capability as a general visual-language retriever. We now analyse how each PreFLMR component affects performance.

[2]The training code of "v2" has not been released officially.

	Model	Vis. Enc.	Text Enc.	Total Param.	I2T WIT R@10	I2T IGLUE R@1	I2T KVQA R@5	Q2T MM R@5	IQ2T OVEN R@5	IQ2T LLaVA R@1	IQ2T Infoseek PR@5	IQ2T E-VQA PR@5	IQ2T OK-VQA PR@5	A.R.
	CLIP				28.1	44.1	23.8	-	22.0	33.0	17.1	10.4	5.7	
	SOTA				FLMR	GIVL	FLMR	ColBERT	FLMR	FLMR	FLMR	Lens	FLMR	
	Res.				23.8	30.8	31.9	86.9	40.5	56.4	47.1	62.5[3]	68.1	
Multi-task Performance														
1	PreFLMR	B	B-v1	207M	41.5	56.8	28.6	77.9	45.9	67.4	48.9	65.4	67.2	9.0
2	PreFLMR	B	B-v2	207M	41.7	57.3	28.6	79.5	46.3	67.2	48.8	67.9	66.1	8.2
3	*w/o inter.*	B	B-v2	207M	41.2	56.8	26.5	78.2	43.7	65.0	47.0	57.3	65.1	10.9
4	PreFLMR	L	B-v1	422M	58.2	69.8	40.6	72.1	59.3	69.3	57.4	70.7	67.9	5.6
5	PreFLMR	L	B-v2	422M	60.5	69.2	**43.6**	78.7	59.8	71.8	57.9	70.8	68.5	3.2
6	*ViT trainable*	L	B-v2	422M	18.7	1.5	0.8	76.7	5.6	54.6	36.7	57.2	58.9	12.3
7	*w/o instruct.*	L	B-v2	422M	13.3	10.5	38.2	75.2	52.1	62.1	49.1	71.3	65.7	9.2
8	*w/o inter.*	L	B-v2	422M	60.0	72.0	40.5	**80.3**	56.1	70.5	55.4	67.0	66.6	4.6
9	PreFLMR	L	S-v1	334M	54.2	66.3	37.9	73.6	53.9	66.0	52.6	66.8	65.3	8.3
10	PreFLMR	L	M-v1	348M	56.2	67.9	37.1	72.9	55.5	64.7	52.2	70.4	65.3	8.2
11	PreFLMR	L	L-v1	677M	49.9	62.8	40.0	72.8	58.8	69.3	59.4	58.2	68.6	6.6
12	PreFLMR	H	B-v2	750M	60.5	71.2	39.4	78.5	61.5	72.3	**59.5**	71.7	68.1	3.1
13	PreFLMR	G	B-v2	1.96B	**61.5**	**71.5**	42.1	78.6	**63.4**	**72.4**	**59.6**	**73.1**	**68.6**	**1.6**
Fine-tuned PreFLMR for Specific Downstream Tasks														
14	PreFLMR	L	B-v2	422M	68.5				70.8		60.3	71.4	67.3	
15	PreFLMR	H	B-v2	750M	69.3				72.3		62.3	72.1	70.5	
16	PreFLMR	G	B-v2	1.96B	69.3				73.1		62.1	73.7	70.9	

Table 6.3 PreFLMR performance on all datasets. PR stands for Pseudo Recall. Best multi-task performance is in bold and best fine-tuning performance on downstream tasks is underlined. For the vision encoder, we compare ViT-B (B), ViT-L (L), ViT-H (H) and ViT-G (G). For the text encoder, we compare Base-v1 (B-v1), Base-v2 (B-v2), Small-v1 (S-v1), Medium-v1 (M-v1), and Large-v1 (L-v1). A.R.: Average Rank against all other models on all tasks. For baselines, we show: GIVL [352] for IGLUE; ColBERTv2 for MSMARCO (MM); FLMR [188] for Infoseek and OK-VQA; and Google Lens [93] for E-VQA. We follow the procedure as detailed in the Appendix C of the E-VQA paper [219] to use CLIP as a zero-shot retriever.

Vision Encoder Scaling

Scaling ViT from ViT-B (86M) to ViT-G (1.8B) while keeping the text encoder fixed brings about substantial performance gain across all tasks (Table 6.3 #2, #5, #12, #13), e.g. 48.8 to 59.6 on Infoseek and 67.9 to 73.1 on E-VQA. The gain is greater when upgrading ViT-B to ViT-L with recall improvements of $\sim 10\%$ on WIT, KVQA, OVEN, and Infoseek, showing the benefit of using better vision encoders. In addition, Fig. B.1 in the appendix illustrates performance gains in scaling the vision encoder with a radar plot. However, the performance plateaus when scaling ViT to H and G. This observation aligns with results reported in the literature. OpenCLIP [50] and BLIP2 [171] have reported marginal or no performance improvement when scaling beyond ViT-L across several datasets. A plausible explanation is that without pre-training on domain-specific data, the ViT model may struggle to distinguish between objects that are visually similar but categorically different in a domain-specific context. For example, the model might find it challenging to differentiate between certain

species of birds, such as the American crow and the common raven, which appear visually similar but belong to distinct ecological niches and exhibit different behaviours.

Text Encoder Scaling

Scaling up the text encoder from BERT-Small-v1 to Medium-v1 to Base-v1 (Table 6.3 #9, #10, #4) yields substantial performance gain (A.R. 8.3, 8.2, and 5.6). However, we find that further scaling to Large-v1 (#11) adversely impacts the performance (A.R. decreased to 6.6). We attribute this to overfitting and unstable training for large models given the available data (Appendix B.2.3). The results suggest that BERT-Base (110M) is adequate for building a capable vision-language retriever.

Improving Text Encoder

Compared to PreFLMR models initialised from Base-v1, models initialised from Base-v2 have better multi-tasking performance indicated by better A.R. (Table 6.3 #1 vs #2 and #4 vs #5). The gain from improving the text encoder is more substantial when using the "ViT-L" vision model (-2.4 A.R.) compared to using "ViT-B" (-0.8 A.R.), indicating that the text encoder is relatively weak as the vision model improves.

6.5.3 Performance of Each PreFLMR Stage

In this section, we analyse intermediate performance in the earlier stages of pre-training to better understand the scaling behaviour of PreFLMR.

Text Encoder Pre-training

Model	MRR@10	Recall@50
Small-v1 (28.8M)	34.5	79.8
Medium-v1 (41.4M)	35.5	81.4
Base-v1 (110M)	35.8	82.4
Large-v1 (345M)	37.0	83.2
Base-v1 (reported in Khattab and Zaharia [146])	36.0	82.9
Base-v2 (reported in Santhanam et al. [264])	39.7	86.8

Table 6.4 Text encoder pre-training results evaluated on the full MS-MARCO test set.

We train "ColBERT-v1" at different sizes and evaluate on the MSMARCO dataset. Table 6.4 shows larger model sizes consistently yield better text retrieval performance. In contrast

to the multi-modal case, scaling up to "Large-v1" does not destabilise training and leads to better performance compared to "Base-v1".

Training the Mapping Structures

	Vis. Enc.	Text Enc.	WIT	LLa.	OVEN	KVQA	IGLUE	Info.	E-VQA	OK.	A.R.
1	ViT-B	Base-v2	34.2	50.9	46.1	28.9	60.5	42.5	32.7	46.5	6.5
2	ViT-L	Small-v1	46.5	46.1	37.9	17.9	57.3	43.5	26.6	56.7	7.0
3	ViT-L	Medium-v1	49.6	47.8	38.6	23.1	58.7	46.7	27.7	**58.1**	5.3
4	ViT-L	Base-v1	49.3	50.8	52.3	38.2	68.5	46.1	41.9	49.4	4.6
5	ViT-L	Base-v2	49.6	51.2	54.8	**40.5**	**69.5**	48.7	**45.0**	50.9	2.3
6	ViT-L	Large-v1	48.5	47.3	51.8	32.8	67.2	45.1	40.0	49.7	5.6
7	ViT-H	Base-v2	**51.8**	51.6	55.3	35.6	69.0	48.6	42.2	51.3	2.8
8	ViT-G	Base-v2	49.5	**51.8**	**59.6**	38.7	69.3	**50.9**	42.4	52.1	**2.0**

Table 6.5 PreFLMR performance after Stage 1. Infoseek, E-VQA, and OK-VQA are tested in zero-shot mode. A.R.: Average Rank against all other models on all tasks. LLa.- LLaVA; Info.- Infoseek; OK. - OK-VQA.

Table 6.5 details system performance after Stage 1 training, in which only the vision-language mapping structure is trained. Similar to Sec. 6.5.2, scaling up the vision encoder improves performance across tasks. PreFLMR exhibits strong zero-shot KB-VQA performance at this preliminary stage (50.87 in Infoseek, 42.44 in E-VQA, and 52.14 in OK-VQA). After Stage 1, PreFLMR with ViT-G performs worse than other variants on IGLUE, E-VQA and OK-VQA. However, it attains the best performance on these datasets after Stage 3. This suggests that tuning the mapping structure alone is not enough to fully utilise larger vision models.

6.5.4 Ablation Studies

Removing Instructions

Removing instructions (Table 6.3 #7) results in much worse overall performance, with the WIT recall rate reduced to 13.3. This shows that instructions are necessary for multi-task learning and that our instruction scheme works well (the full list of instructions is given in Appendix B.1). We observe a slight improvement in performance on E-VQA, from 70.8 to 71.3, after removing instructions. A plausible explanation for this is that the model undergoes extensive training on E-VQA during the intermediate pre-training stage, leading to better fitting to this dataset but not to others.

Pre-training Datasets

Datasets	WIT	LLaVA	Infoseek
All	34.14	50.82	42.71
w/o CC3M	29.33	44.82	40.18
w/o LLaVA	33.78	30.78	39.20
w/o MSMARCO	33.96	47.88	38.90
w/o OVEN&KVQA	33.96	49.85	35.62

Table 6.6 Ablation study on Stage 1 pre-training datasets. The model is ViT-B + Base-v1. We evaluate systems on Infoseek in zero-shot mode though it is not used in Stage 1 training.

As shown in Table 6.6, adding CC3M to training improves performance on all metrics, showing that learning to understand scene via captioning datasets is beneficial. Removing either LLaVA or MSMARCO harms zero-shot KB-VQA performance (-3.0 in Infoseek), noting that Infoseek is not used in this stage. Training on these datasets facilitates learning question-aware visual representations as the cross-attention in the mapping structure must attend to the text input to perform well on these tasks. Omitting knowledge-intensive datasets (OVEN and KVQA) negatively impacts the zero-shot performance on Infoseek, showing the importance of using in-domain data in training the mapping structure.

Mapping Structure Scaling

	N_{TR}	WIT	LLaVA	Infoseek
ViT-B + Base-v1	1L	34.1	50.8	42.7
ViT-B + Base-v1	4L	29.0	51.4	40.8
ViT-L + Base-v2	1L	49.6	51.2	48.7
ViT-L + Base-v2	4L	45.9	51.7	46.8

Table 6.7 Performance of adding more Transformer layers to the mapping structure. N_{TR} is the number of Transformer layers in the mapping structure.

Table 6.7 illustrates the impact of scaling up the mapping structure under two PreFLMR configurations. Increasing cross-attention layers from 1 to 4 marginally improves LLaVA performance (+0.5, approx.), but adversely impacts performance on WIT (-4, approx.) and Infoseek (-2, approx.). We adhere to the 1-layer design, noting that adding parameters to the mapping structure does not improve performance.

[3]The performance is not fully comparable due to differences in the construction of the test passage corpus and the proprietary nature of the data and pipeline used in Lens. The reported figures serve as a reference point.

Intermediate Pre-training

Stage 2 improves performance on KB-VQA tasks (Table 6.3 #3 vs #2 and #8 vs #5). With intermediate pre-training, the score on other KB-VQA tasks (Infoseek, KVQA, OK-VQA) increases by $\sim$1% or more. This shows that E-VQA is an appropriate corpus for intermediate pre-training. We analyse the gain from intermediate pre-training in more detail in Sec. 6.5.6.

6.5.5 Retrieval Augmented Visual Question Answering with PreFLMR

Model	OK-VQA	Infoseek	E-VQA
Baseline	66.10	21.80	48.80
Baseline model	PaLM-E	PALI-X	PaLM-B + Lens
AVIS	60.20	50.70/56.40[4]	-
RA-VQA-v2 w/ FLMR	60.75	-	-
RA-VQA-v2 w/ PreFLMR	61.88	30.65	54.45
w/o retrieval	55.44	21.78	19.80

Table 6.8 Downstream KB-VQA performance when RA-VQA-v2 (Chapter 5) is equipped with PreFLMR and fine-tuned on the target M2KR's KB-VQA sub-tasks. AVIS [117] is a recently published hybrid system that leverages many planning stages to solve KB-VQA questions, which we include for reference.

We build on RA-VQA-v2 proposed in Chapter 5 to tackle OK-VQA, Infoseek, and E-VQA. We fine-tune the best-performing PreFLMR variant on the target retrieval task (ViT-G + Base-v2, Table 6.3 #14) and follow RA-VQA-v2 to fine-tune a BLIP2 answer generator (Sec. 5.3.2) on the target M2KR KB-VQA task.[5] Following previous literature [273, 44, 219], we use VQA score, Accuracy, and BERT matching scores (BEM) [25] to evaluate performance on OK-VQA, Infoseek, and E-VQA, respectively.

A brief summary of the systems shown in Table 6.8: PaLM-E [73], PALI-X [40] and PaLM-B [10] are large multi-modal models with 562B, 55B, and 1T parameters, respectively. The E-VQA state-of-the-art (as of date) [219] uses Lens [93], the Google API for image retrieval. AVIS [117] is a hybrid system with many components (such as PaLI, PaLM, and Google Lens&Web Search API) and planning stages powered by LLMs. We note that

[4] 50.7 for Unseen Entity and 56.4 for Unseen Question; no overall accuracy is reported.

[5] We note that this work was conducted during the early stage of the release of Infoseek and E-VQA. We prepared the data splits according to the need for retrieval training following Appendix B.1. The systems are trained and evaluated on the data splits provided in M2KR to show the improvement relative to systems without retrieval.

PreFLMR could be used as part of the AVIS pipeline to enhance its ability to fetch relevant documents given questions and images.

As shown in Table 6.8, compared to models without retrieval, PreFLMR improves performance by approximately 6% on OK-VQA, 9% on Infoseek, and 34% on E-VQA. These results highlight the effectiveness of PreFLMR in document retrieval for KB-VQA tasks.

On OK-VQA, the performances of RA-VQA-v2 (PreFLMR) and RA-VQA-v2 (FLMR) are similar. Table 6.3 #13 shows that PreFLMR attains similar Recall@5 as FLMR on OK-VQA even though it has a much larger vision encoder. As a possible explanation, compared to E-VQA and Infoseek, the knowledge required to answer OK-VQA question is less specialised and many OK-VQA questions can be answered without document retrieval [219]. See Appendix B.5 for qualitative analysis that compares OK-VQA and E-VQA questions. Another possibility is that, compared to E-VQA and Infoseek where the ground-truth document is provided for each question, the OK-VQA training set does not provide ground-truth knowledge documents. The retriever uses pseudo-relevant documents in training that contain the target answer but these may not be truly useful for answering the question. This is evidence that data quality should also be improved along with model scaling.

6.5.6 Analysis of Intermediate Pre-training

Sec. 6.5.4 shows that Stage 2 Intermediate Pre-training improves the performance as evaluated by task-specific metrics. In this section, we further quantify the gains from Stage 2 for each dataset and more clearly show that KB-VQA tasks benefit more from Stage 2 than other tasks. We use the difference in minimal validation loss[6] achieved on each dataset starting from checkpoints before or after Stage 2 Intermediate Pre-training as a measure of benefit. This enables comparison of tasks with different performance metrics. Intuitively, a larger absolute difference in validation loss indicates that the dataset benefits more from the Intermediate Pre-training stage.

Figure 6.2 plots the difference in validation loss of every dataset when the starting checkpoints have undergone N_{inter} intermediate pre-training steps using either BERT-medium or BERT-base as the text encoder backbone. As expected, starting from E-VQA-pre-trained checkpoints yields lower validation loss in knowledge-intensive tasks such as OK-VQA, KVQA, and OVEN after the same number (5,000) of fine-tuning steps. Performance on these datasets indeed sees more gain from Stage 2 training (Table 6.3, #5 v.s. #8). Figure 6.2 also indicates the existence of an optimal N_{inter}, beyond which the model overfits to

[6]We find that the validation loss is predictive of the actual performance. A lower validation loss usually suggests a better performance in the tasks that we study.

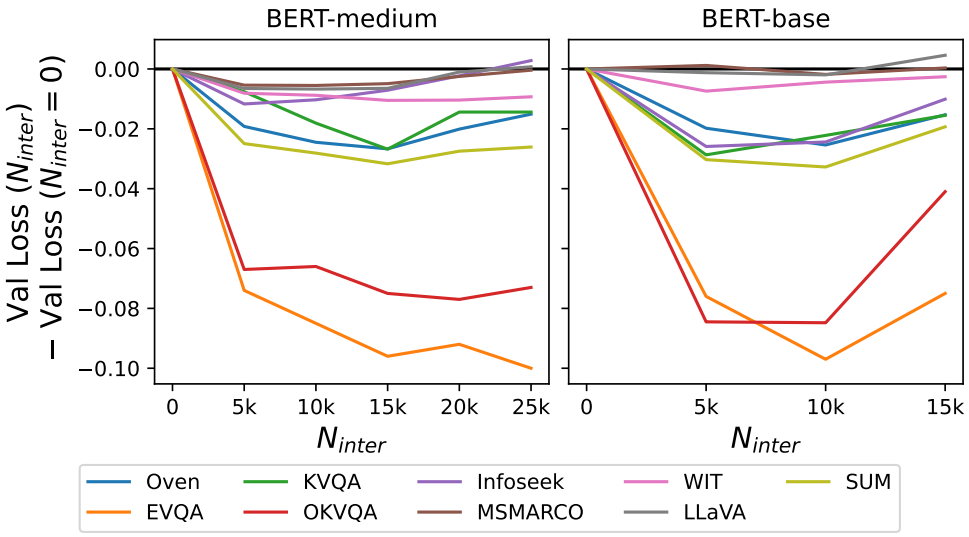


Fig. 6.2 Change in Stage 3 validation loss when initialised from Stage 2 checkpoints after N_{inter} steps of intermediate pre-training. A large difference indicates a greater gain from intermediate pre-training.

E-VQA, harming performance on other datasets. The larger PreFLMR model with BERT-base text encoder overfits faster than PreFLMR with BERT-medium ($N_{inter} \approx 15,000$ versus $N_{inter} \approx 10,000$). We use V-Entropy [335] to formalise our analysis as an empirical measure of mutual information between datasets in Appendix B.4.

6.5.7 Summary of Findings

We summarise the results of our investigations into scaling behaviour as follows:

- The text encoder size need not exceed that of BERT-base (110M) to achieve competitive multi-modal retrieval performance (Sec.6.5.2).
- Scaling up the vision encoder from ViT-B to ViT-G yields substantial gains (Sec.6.5.2).
- Scaling up the mapping structure does not improve performance (Sec.6.5.4).
- Intermediate pre-training on high-quality in-domain data (E-VQA) effectively improves retrieval performance across KB-VQA tasks (Sec.6.5.4, 6.5.6).
- Strong knowledge retrievers boost performance on challenging KB-VQA tasks such as OK-VQA, Infoseek, and E-VQA via Retrieval-Augmented Generation (Sec.6.5.5).
- Ground-truth document labels are important to make full use of large models in training multi-modal retrievers (Sec.6.5.5).

Limitations and Potential Future Work

Limited by available computational resources, we leave several further investigations as future work: (1) The CLIP-ViT models [50] were not pre-trained on in-domain data of knowledge-intensive tasks. Further training may enhance the model's ability to recognise a broader range of objects; (2) Advanced training approaches beyond contrastive learning, such as score distillation [264], could be explored to further enhance retrieval performance; (3) Investigating a more optimal mix proportion of datasets with varying sizes also warrants further exploration.

6.6 Summary

This chapter has studied the scaling of state of the art multi-modal document retrieval systems, with a focus on enhancing fine-grained late-interaction retrieval for knowledge-based visual question answering. We contribute a comprehensive training and evaluation framework, M2KR, for general-purpose multi-modal knowledge retrieval. The PreFLMR system we train in the M2KR framework yields excellent retrieval performance across a range of tasks and can also serve as a base for further task-specific fine-tuning.

Additionally, the work discussed in this chapter addresses research questions RQ1 and RQ2, details of which will be elaborated in the final chapter of the thesis (Sec. 9.1).

In the next two chapters, we shift the focus to another type of knowledge-intensive question answering - Table Question Answering (TableQA). We will present two TableQA systems that are deeply integrated with retrieval models, which attain state-of-the-art performance on popular TableQA tasks.

Chapter 7

An Inner Table Retriever for Robust Table Question Answering

7.1 Introduction

In the preceding chapters, we examined the methodologies utilised in KB-VQA, where we presented retrieval methods as substantial improvements for KB-VQA systems. Beginning with this chapter, our focus shifts to exploring retrieval methods for TableQA. This new endeavor aims to further verify the effectiveness of retrieval methods in a different context.

Tables offer a systematic way of storing information in the Web and elsewhere. Extracting information from Web tables poses different challenges than extracting information from relational databases with logical queries, especially when queried via Natural Language (NL) user questions. Table Question Answering (TableQA) is the task of answering such questions with answers extracted from table content. This requires developing models with the ability to reason over and understand tables. TableQA has broad applications across various domains. For instance, TableQA can address user inquiries about product properties, like specifications and prices, stored in tables on online shopping platforms. Additionally, TableQA is capable of extracting specific data, such as numbers from a datasheet, in email attachments as requested by users. In this thesis, we focus on relational tables, where data is structured into rows and columns. Each table represents a type of entity, and each row corresponds to a specific instance of that entity. Columns represent attributes of the entity. For instance, considering the previous example of product properties, such information could be structured into a table with rows labelled ‘price: 1,000 pounds’ and ‘length: 3 meters’, with ‘price’ and ‘length’ are two properties.

Following the success of the pre-training paradigm for understanding NL text [66], some recent research has focused on pre-training Transformer models [307] on large corpora of *linearised* tables in a self-supervised fashion using encoder-only architectures [106, 353, 339], or encoder-decoder architectures [200, 135]. These so-called Tabular Language Models (TaLMs) [69] were fine-tuned on the TableQA downstream task—among others—to achieve state-of-the-art performance [106, 200]. However, the self-attention mechanism in TaLMs has a quadratic complexity on the dimensionality of the input and tables, which might consist of tens or even hundreds of rows and columns, thus yielding longer sequences than the TaLMs can easily handle. State-of-the-art TableQA models handle this limitation by truncating the linearised table to fit an input length budget, e.g., of 512 and 1024 tokens by Herzig et al. [106, TaPas] and Liu et al. [200, TaPEx], respectively. In other applications, simple sequence truncation might be reasonable, e.g., encoding only the initial paragraphs of a Wikipedia document presuming it comprises a summary, or dropping earlier turns in Conversational QA to focus on the recent ones. However, in TableQA it is not realistic to assume that relevance depends on the position within the linearised sequence, especially because different questions require various table regions to be properly answered. For example, even for a standard data set as WikiTableQuestions, naive truncation allows information loss affecting 18.1%-44.9% of tables, which limits QA accuracy (see Sec. 7.4.1 for details). This is also an important limitation in latency-constrained realistic use cases that use big tables. To this end, a content-driven strategy is needed to avoid information loss. We refer to tables exceeding the input length budget as ***overflow*** tables, as opposed to ***compact*** tables, which fit within the budget. An example of an overflow table is shown in Fig. 7.1(a), where naive truncation leads to the wrong answer.

In this chapter, we propose a novel retrieval methodology, ITR (Inner Table Retriever), to improve on this problem by creating smaller sub-tables, i.e., within a length budget, based on dense retrieval of table rows and columns according to the relevance to the question. An example is shown in Fig. 7.1(b). Our method is flexible and can be integrated *off-the-shelf* into virtually any existing TableQA system. To the best of our knowledge, our work is the first to propose a sub-table selection strategy based on neural models in the context of TableQA that improves the accuracy, especially for the overflow tables, setting a new state of the art. Other input selection strategies, mainly heuristics-based, have also been proposed in the literature [152, 353, 78], which we discuss further in Sec. 7.2.

We aim to develop a pipeline designed to accommodate any TableQA models based on two key design rationales. First, recent work introduces new Transformer architectures to process long tables, which typically require retraining the models and could potentially degrade performance compared to full self-attention Transformers [78, 152]. Second, it

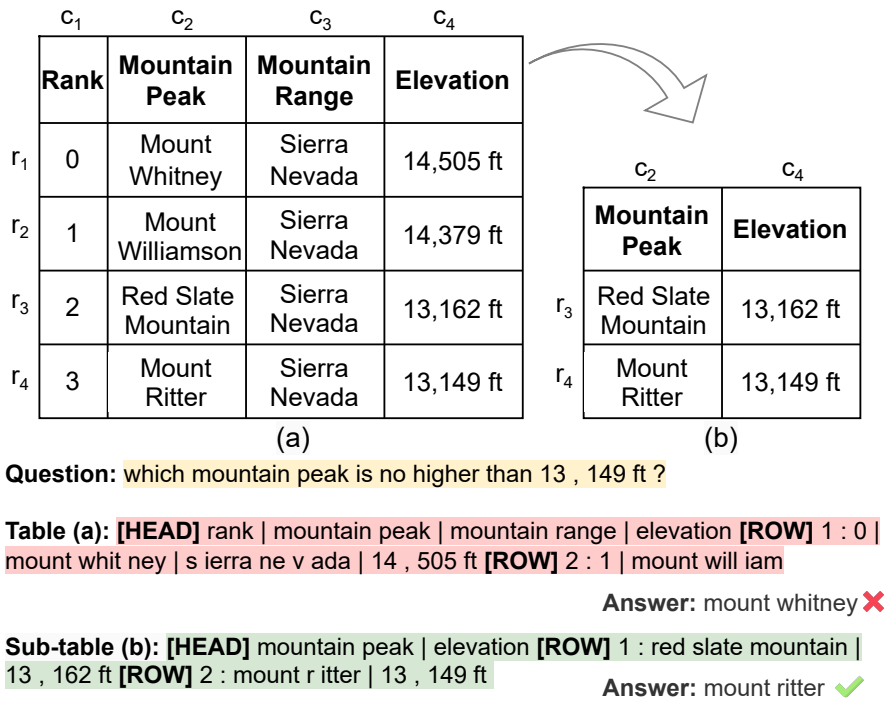


Fig. 7.1 TableQA example with the model input length budget set to 50 tokens using TaPEx tokenisation and table linearisation format; (a) is an *overflow* table because the linearised version must be truncated. Our method can identify sub-tables like (b) within the length budget, removing the information loss.

is crucial that our pipeline is versatile enough to support any TableQA models, including those specifically designed for handling long tables. This versatility is important because integrating retrieval methods is expected to effectively minimise irrelevant and redundant information, thereby enhancing overall model performance even for models that is already able to process very long tables.

To summarise, the contributions of this chapter are the following:

1. We propose ITR, an efficient approach to handling overflow tables for TableQA models, which produces sub-tables containing the most relevant information for answering a question while fitting within the budget.

2. We combine ITR with existing TableQA systems such as TaPas, TaPEx, and OmniTab [135], and achieve a new state of the art result for two standard benchmarks, WikiSQL and WikiTableQuestions.

3. We evaluate the robustness of ITR against current TableQA models on *overflow* tables (defined earlier in this section), when reducing the length budget, and when repositioning relevant table information.

The code and data have been released at: https://github.com/amazon-science/robust-tableqa.

7.2 Related Work

This section provides an overview of the related work up to January, 2023, the time of this study.

The works most related to ours employed different pruning strategies to handle large tables. Yin et al. [353, TaBERT] introduce the concept of *content snapshot* as encoder input. This snapshot is composed of a small number of rows which are chosen based on the n-gram similarity between the question and column headers and cell values. In a similar fashion, Eisenschlos et al. [77] explore Jaccard similarity to obtain the most similar columns. In addition, they leverage model tokenizer to reduce cell tokens to their first token, when necessary, and dropping entire rows that do not fit the length budget. However, lexical similarity and naive truncation are not optimal and may lead to information loss, which has a drastic effect on TableQA performance, as we show later in our experiments (Sec. 7.5) and in example system outputs (Sec. 7.7).

Another line of work focuses on balancing model efficiency and accuracy when handling long tables. Krichene et al. [152, DOT] first uses a smaller *pruning* transformer to selects top-K tokens from the input table, and then a larger second task-specific transformer takes into consideration only the selected K tokens and their pruning scores; Eisenschlos et al. [78, MATE] can accept more tokens while not significantly increasing latency. Authors apply sparse self-attention and use different attention heads for tokens in the same column and row. However, their proposed mechanisms are intricately embedded within the model, necessitating the model's retraining to ensure proper functionality. In contrast, our ITR method, as a flexible plug-in process, can work independently of any underlying TableQA model without retraining. Moreover, our approach is complementary to theirs since ITR can drop irrelevant information in tables efficiently and pass the trimmed compact table to the underlying model. This can further exploit the potentials of virtually any TableQA models and improves their performance.

Although not specific to TableQA, works such as Wang et al. [315] and Chen et al. [41] employ chunking strategies, i.e., encoding table chunks separately and then aggregate them together. However, chunking is not widely employed in the literature [69], due to encoding overhead requiring multiple inference calls for each chunk. This additional computational

demand can significantly slow down the processing time and increase resource usage, making it less efficient compared to single-pass methods that encode the entire table at once. In contrast, our proposed solution chooses the most relevant table content and encode only once.

7.3 Method

7.3.1 Task

Given a question q and a table T, TableQA systems return an answer denotation a, either by performing table cell selection or as the result of operations (such as counting) carried out over an aggregation of table cells. As defined in Sec. 2.5, 'denotation' refers to a list of cell values or numerical values derived from selected table regions via aggregation functions (e.g., SUM, which adds up the values in the selected cells) [237]. Our approach aims to find one or more sub-tables T_{sub} containing the most relevant information from T needed to answer q; T_{sub} can replace T as the input to virtually any existing TableQA system.

To this end, we map a table T into a set of items, where an item is either a complete row or a complete column. For an $n \times m$ table, this gives a set of items $\{r_1, \ldots, r_n, c_1, \ldots, c_m\}$. Then, we construct sub-tables by specifying subsets of these rows and columns: a sub-table consists of the cells at the intersection of the selected rows and columns. We refer to each such set of rows and columns as a *mix*, and note that a mix must contain at least one row and one column to specify a valid sub-table. The table in Fig. 7.1(a) is defined as $\{r_1, r_2, r_3, r_4, c_1, c_2, c_3, c_4\}$, and the sub-table in Fig. 7.1(b) is specified as the mix $\{r_3, r_4, c_2, c_4\}$. $\{c_2, c_4\}$ is not a valid sub-table, since no cells will be intersected with only column-wise items.

7.3.2 Inner Table Retriever

ITR is a process of retrieving table rows and columns, and combining them to form sub-tables T_{sub}, with q as a query. We describe the steps for creating sub-tables in Algorithm 1. Lines 2-6 compute item similarities to q, and the function `Items`(T) in Line 4 maps T into its $n+m$ items. Following Karpukhin et al. [143, DPR], we have two fine-tuned encoders, one for questions (E_Q) and another for table items (E_T).

We fine-tune DPR encoders using a question as a query and the row/columns that contain the gold answer cells as positive items. Then we sample the negative items from the remaining row/columns in the table. We leverage the standard DPR contrastive loss (as used in Sec. 3 Eq. 3.2) to fine-tune the two encoders so that the similarities between question and positive

Algorithm 1 Creating N sub-tables from T for q.

```
1:  def ITR(q, T, E_Q, E_T, N, b):
2:      Z ← [], T_sub ← [], T_aux ← [], L ← []
3:      e_q ← E_Q(q)
4:      for each i ∈ Items(T) do
5:          e_t ← E_T(i)
6:          Z ← Z ∪ {sim(e_q, e_t), i}
7:      for each i ∈ Sorted(Z) do                          ▷ ↓ sim
8:          T_aux ← T_aux ∪ i
9:          if CheckValid(T_aux) then
10:             T_sub ← T_sub ∪ T_aux
11:     for each t ∈ T_sub do
12:         L ← L ∪ {Length(t), t}
13:     T_sub ← []
14:     for each (l, t) ∈ Sorted(L) do                     ▷ ↓ length
15:         if l > b then continue
16:         T_sub ← T_sub ∪ t
17:     return T_sub[ : N]
```

item embeddings are maximised.[1] At inference time, we compute the contextual embeddings for the question (Line 3) and all the items in a table (Line 5) and compute their similarity, sim$(\cdot)$, as the dot product in Line 6. In practice, pre-computed embeddings of table items can be cached offline.

Creating sub-tables

In Lines 7-10 we loop through the table items ranked by highest similarity and aggregate in T_{aux}. `CheckValid`(T_{aux}) verifies that T_{aux} is a valid sub-table, i.e., there exists at least one column-row intersection (see Sec. 7.3.1).

Choosing the most appropriate sub-tables

In Lines 11-16 we sort the sub-tables by their sequence length in descending order, and filter out any sub-table which exceeds the length budget b. Finally, in Line 17 we return the top-N remaining largest sub-tables that fit the budget length to be used as input to the TableQA model. Each returned sub-table is guaranteed to contain the most relevant items, due to the sorting operation in Line 7.

[1] More details in Appendix C.1.1.

7.3.3 TableQA with ITR

Through the ITR process, we obtain N sub-tables T_{sub} as replacement for the original table T. Each sub-table, together with the associated NL question q, is linearised into a sequence of tokens prior to encoding, using the corresponding TableQA tokenizer. We exemplify this in Fig. 7.1, where both table (a) and sub-table (b) are similarly processed. Each linearised sequence is used as input to the TableQA model, thus obtaining N predictions, out of which we choose the most confident answer. We empirically find that the model performance is marginally better when $N > 1$, instead of only considering the largest sub-table (see Sec. 7.6; Appendix C.3).

7.4 Experimental Setup

7.4.1 Datasets and Evaluation

We use two popular datasets for TableQA which are constructed from Wikipedia: WikiSQL[2] [381] and WikiTableQuestions[3] [237, WikiTQ], with each posing different challenges. WikiSQL is a simpler TableQA dataset than WikiTQ as it requires mainly filtering and simple operations of table cells to answer a question. WikiTQ demands more complicated reasoning capabilities such as aggregations, comparisons, superlatives, or arithmetic operations, e.g., SUM, MAX. We measure DA (Denotation Accuracy) for validation and test sets, to assess whether the predicted answer(s) equals the ground-truth answer(s). Additionally, we introduce the distinction between *compact* tables and *overflow* tables, which is determined by the length of linearised question-table pair. Statistics in Table 7.1 show that even when using a relatively high number of tokens, i.e., 1024 or 512—the allowed maximum supported by most of Transformer-based encoders— the range of overflow tables is very high, 7-33% in WikiSQL, and 18-45% in WikiTQ. ITR is applied only for overflow tables, as compact tables already fit in the token budget. Finally, we evaluate ITR retrieval ability on WikiSQL using Recall@K, which measures whether *all* the gold rows/columns for answering a question are among the top-K retrieved items.

[2]https://huggingface.co/datasets/wikisql
[3]https://huggingface.co/datasets/wikitablequestions

	WikiSQL		WikiTQ	
max tokens	Dev	Test	Dev	Test
1024	6.8	9.7	19.2	18.1
512	30.6	32.7	44.4	44.9
256	68.6	69.9	81.5	82.6
128	98.0	98.2	100	100
64	100	100	100	100
total samples	8421	15 878	2831	4344

Table 7.1 Total number of samples and overflow rate (%) for different length budgets in WikiSQL and WikiTQ. Question-table pair sequence length is calculated based on TaPEx's tokenizer and linearisation strategy.

7.4.2 Training Setup

For both ITR retrieval component and the underlying TableQA systems that we train in-house, we choose the best checkpoint based on the performance on the validation set. Otherwise, for TableQA systems in the literature, we use the released checkpoints from `huggingface`.

Since WikiTQ does not provide SQL annotations, which are needed to obtain gold answer cell coordinates for supervising the ITR retriever (see Sec. 7.3.2), we use the model trained on WikiSQL in a zero-shot fashion to retrieve the relevant table items for the WikiTQ TableQA task. We set the number of sub-tables N=10 when using TaPEx and OmniTab systems, and N=1 with TaPas (see Appendix C.3). We provide details and hyperparameters for ITR and TableQA models in Appendix C.1.

7.4.3 Comparison Systems

We evaluate the effectiveness of ITR by comparing our ITR-enhanced baselines with recent models from literature that report performance on WikiSQL or WikiTQ [221, 106, 358, 200, 135]. We provide detailed comparisons using TaPEx, TaPas, and OmniTab, with ITR included in inference alone, as well with ITR integrated into TaPEx training. TaPEx (Sec. 2.5, Fig. 2.47) leverages BART [166], an encoder-decoder model, as the base model. TaPEx takes as input a <question, table> pair and autoregressively generates an answer, implicitly performing any kind of aggregations. OmniTab extends TaPEx and leverages multi-tasking and data augmentation to establish a new state of the art in WikiTQ. TaPas is an encoder model built on top of BERT [66] which takes as input a <question, table> pair and learns jointly to: i) select table cells by thresholding the cell confidence, and ii) predict an explicit operator to aggregate results from the cell selection. We use a recent version of TaPas

Models	Dev	Test
Min et al. [221]	84.4	83.9
TaPas$_{v0}$ [106]	85.1	83.6
TaPas [77]	89.8	-
Yu et al. [358]	85.9	84.7
TaPEx [200]	89.2	89.5
OmniTab [135]	-	88.7
ITR $\rightarrow$ TaPEx (#6)	91.8	91.6
ITR $\rightarrow$ TaPas (#5)	**92.1**	**92.1**

Table 7.2 Results on WikiSQL. **Bold** denotes the best DA for each split. # references a row in Table 7.4.

proposed by Eisenschlos et al. [77] which leverages intermediate pre-training and table pruning, e.g., cell truncation to the first token and dropping rows that exceed the limit, improving significantly the initially released model (TaPas$_{v0}$). When applying ITR, we disable TaPas table pruning and use the full sub-table(s). Regarding their capacity, TaPEx and OmniTab support up to 1024 tokens, while TaPas can take up to 512 tokens.

It is worth noting that we have not been able to fully reproduce the reported results in the literature of TaPas, TaPEx and OmniTab with their `huggingface` implementations. This discrepancy is attributed to cross-framework dependencies, as well as model-specific preprocessing and evaluation scripts. To assess the realistic contribution of ITR, we also report reproduced results using the same unified framework across all models. We discuss reproducibility in Appendix C.1.3.

7.5 Results

7.5.1 Main Results

We report the performance of our best ITR-enhanced TableQA models for WikiSQL and WikiTQ in Tables 7.2 and 7.3, respectively, and compare with the state-of-the-art results as reported in the literature. In WikiSQL, ITR-enhanced models consistently outperform all previous baselines. ITR improves TaPEx and TaPas performance with 2.6 and 2.3 DA points in the Dev set, respectively. ITR $\rightarrow$ TaPas sets a new state of the art for WikiSQL, reaching a DA of 92.1% in the Test set. In WikiTQ, results are mixed: ITR shows slight degradation over TaPas in the Dev set. Combined with TaPEx, ITR improves DA by 4 points. Further,

Models	Dev	Test
TaPas$_{v0}$ [106]	29.0	48.8
TaPas [77]	50.9	-
Yin et al. [353]	53.0	52.3
Yu et al. [358]	51.9	52.7
TaPEx [200]	57.0	57.5
OmniTab [135]	-	62.8
ITR → TaPEx (#9)	61.8	61.5
ITR → TaPas (#5)	50.5	50.8
ITR → OmniTab (#7)	**62.1**	**63.4**

Table 7.3 Results on WikiTQ. **Bold** denotes the best DA for each split. # references a row in Table 7.4.

		WikiSQL				WikiTQ			
#	Models	Dev	Test	Compact	Overflow	Dev	Test	Compact	Overflow
1	TaPas (`hf`)	90.4	89.5	92.1	76.3	50.2	50.6	55.4	37.6
2	TaPEx (`hf`)	89.5	88.7	91.9	59.1	57.2	55.5	60.5	32.7
3	OmniTab (`hf`)	-	-	-	-	61.0	62.1	**66.5**	35.9
4	TaPEx	88.4	87.7	90.8	58.2	58.7	57.8	62.8	35.3
5	ITR → TaPas (`hf`)	**92.1**	**92.1**	**92.1**	**92.1**	50.5	50.8	55.4	38.2
6	ITR → TaPEx (`hf`)	91.8	91.6	91.9	89.1	58.4	56.9	60.5	41.1
7	ITR → OmniTab (`hf`)	-	-	-	-	**62.1**	**63.4**	**66.5**	**45.3**
8	ITR → TaPEx	90.5	90.6	90.8	87.7	60.4	58.8	62.8	41.1
9	ITR → TaPEx *(+train)*	91.3	91.4	91.6	89.4	61.8	61.5	65.2	44.8

Table 7.4 DA on WikiSQL and WikiTQ when applying ITR at inference time only or also in training. TaPEx denotes the in-house fine-tuned TaPEx (`hf`) model. Compact and Overflow are subsets of the Test split. Only Dev/Test DA values are directly comparable across models. This is because the token limit is different for TaPEx and OmniTab (1024) and TaPas (512), and overflow samples are of different sizes for these models, i.e., 9.7% for TaPEx and OmniTab (see Table 7.1) and 16.6% for TaPas. **Bold** denotes the best accuracies for each dataset split.

combined with OmniTab, ITR improves DA by 0.6%, reaching a new state-of-the-art result on this task.

In Table 7.4 we compare ITR in a unified experimentation setting using the released model checkpoints in `huggingface` for inference-only evaluation, and the in-house fine-tuned TaPEx model that enables both training and evaluation with ITR. For OmniTab, only the WikiTQ model is made available, thus we do not evaluate in WikiSQL. We also break

down the performance into compact and overflow tables to better assess the contribution of ITR. When evaluated using the same settings, ITR consistently improves on top of baselines (#5-7 *versus* #1-3) on WikiSQL and WikiTQ, respectively: TaPas (+2.6% and +0.2%), TaPEx (+2.9% and +1.4%) and OmniTab (+1.3% on WikiTQ only). This is also true for the in-house trained TaPEx (#4 *versus* #8). ITR reduces the compact/overflow performance gap significantly: ITR increases overflow DA with up to 30% (with TaPEx) and 9.4% (with OmniTab) on WikiSQL and WikiTQ, respectively, making the underlying model more robust to larger tables. In fact, in WikiSQL we almost close the gap between compact and overflow tables (#5). These results show that ITR can be applied at inference time to always improve performance by presenting more complete and relevant information in state-of-the-art model decision process, even without interfering with the model training process.

Additionally, fine-tuning TaPEx with ITR sub-tables (#9) is better than using ITR only at inference time (#8). Training with ITR improves the *overflow* DA by 1.7% on WikiSQL and 3.7% on WikiTQ. The Test performance increases by 0.8% and 2.7%, respectively. Not only does ITR improve the decision process of underlying models at inference time, but it further increases performance if included in the model training process as well. This demonstrates that ITR can be flexibly applied to any TableQA model.

7.5.2 Repositioning Denotations

Models	Test	Test$_{ext.}$	Compact	Compact $_{ext.}$	Overflow	Overflow$_{ext.}$
TaPEx	87.7	82.8	90.8	89.1	58.2	23.6
ITR → TaPEx	91.4	90.0	91.6	90.5	89.4	85.3
ITR → TaPEx *(+train shuffled)*	91.5	**90.8**	91.8	**91.2**	89.0	**87.1**

Table 7.5 DA in WikiSQL when repositioning denotations to the bottom-right corner of the table, denoted as D$_{ext.}$ where D is any of the dataset splits. **Bold** denotes the best DA for each dataset split in the extreme scenario.

In Table 7.4 we notice that models that naively truncate table sequences (#1-4) may observe the correct denotations within the length budget, achieving 58.2-63.2% DA in WikiSQL even for overflow samples. This is because the original table happens to present the answer early, e.g., in the first column/row. To fully investigate the potential and usefulness of ITR, we create an extreme case by moving gold rows and columns altogether to the bottom right of each table, thus reducing the chances of arbitrary success due to dataset design. For this analysis, we use WikiSQL since it provides the gold cell annotations. We evaluate TaPEx in combination with ITR on both the original and extreme-case set and report results in Table 7.5.

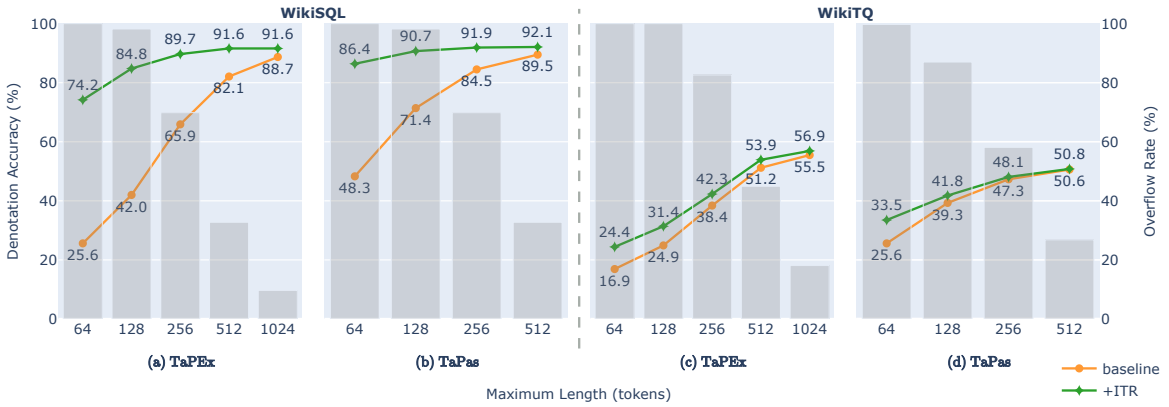


Fig. 7.2 Impact of input length budget on Denotation Accuracy (line plots) and Overflow Rate (bar plots) for WikiSQL (left) and WikiTQ (right) Test sets.

Unsurprisingly, the overall performance of TaPEx drops sharply from 87.7% to 82.8%, with the overflow accuracy dropping 34.6%, from 58.2% to only 23.6%. This suggests that in extreme cases failure to address the long table issue will harshly degrade model performance and that the reported DA values can be optimistic due to the convenient position of the relevant information at the top of the table. ITR-enhanced TaPEx is less affected: the overall DA degrades by 1.4% and overflow accuracy by only 4.1% as opposed to 34.6% drop of TaPEx. The 4.1% drop is mainly due to the positioning bias that the TableQA system might have learned during training. In fact, if we introduce row/column repositioning also at training time *(+train shuffled)*, i.e., making gold answer denotations appear equally in any possible position of the input sub-table, we observe a reduced impact of the extreme repositioning at inference time: the performance drop is only 0.7% overall and 1.9% in the overflow split. We further discuss row/column positioning effects in Appendix C.2.

7.5.3 Reducing the Input Length Budget

In production pipelines, latency is a crucial issue. Minimising the number of tokens at the model input, and consequently the overall latency, can significantly enhance user experience. We explore the input length budget within 64 to 1024 tokens and compare TaPEx and TaPas when combined with ITR in WikiSQL and WikiTQ.

We recall from Table 7.1 that overflow rate increases drastically with shorter length budgets. TaPEx and TaPas use different linearisation and tokenization strategies, thus they yield different overflow rates for the same budget. However, for both models and dataset, 64 tokens leads to an 100% overflow rate.

We show the DA values in WikiSQL and WikiTQ in Fig. 7.2. In WikiSQL, ITR helps the TableQA model remain in the region of the best accuracy even when reducing the budget to 256 and 128 tokens for TaPEx and TaPas, respectively. Without the aid of ITR, both models sharply degrade in performance. Interestingly, TaPEx-based models are less robust when reducing the number of tokens than the TaPas-based model. This is because TaPEx linearisation uses up tokens faster, therefore encoding less table information overall, and TaPas employs a naive truncation strategy to fit cells as much as possible even at the cost of truncating cell contents (see Sec. 7.7 for examples of system outputs). In WikiTQ, while the trend remains unchanged, there is a more noticeable drop for all the models. This is due to the challenging nature of WikiTQ questions which generally require visibility of larger portion of tables. However, ITR still improves over baselines, further widening the gap as we decrease the length budget to 128 and 64 tokens. Similarly to results in WikiSQL, TaPas is more robust in extreme scenarios in WikiTQ.

Thus, while ITR benefits models in standard benchmarks, it is even more beneficial in extreme realistic scenarios.

7.6 Ablation Study

7.6.1 ITR Variants

In addition to our ITR, we investigate several variants: with varying number of sub-tables N, representing tables using columns or rows only, creating sub-tables with different strategies, or scoring items via a different measure of relevance.

Row/Column-only Items. Our ITR considers a mix of both row and column items. We redefine Algorithm 1 to consider only row or column items, but not both, creating the following ITR variants:

1. ITR$_{col}$: $\texttt{Items}(T)$ maps a table T into a set of columns, e.g., $\{c_1, c_2, c_3, c_4\}$ in Fig. 7.1(a). A sub-table is created by combining the retrieved columns, e.g., Fig. 7.1(b) would be represented as $\{c_1, c_3\}$ but containing all 4 rows for the 2 columns.
2. ITR$_{row}$: similar to ITR$_{col}$, but only considering row-wise items.

Reduction *versus* Addition. ITR returns the largest N possible sub-tables by iteratively dropping irrelevant items to fully leverage the length budget (here referred to as *Reduction*, and models are suffixed by '$^-$'). As such, we do not consider the remaining smaller sub-tables. To verify whether dropping the irrelevant items is a better approach, we contrast it

with an *Addition* strategy (models suffixed by '$^+$'), where we return the top-N sub-tables created by successively appending the top-N items by their similarity with q, i.e., after Line 10 in Algorithm 1.

Semantic versus lexical. Finally, inspired by table input selection strategies in literature [77, 353], we use n-gram similarity instead of dense retrieval in Algorithm 1 (Lines 5-6) and obtain ITR$_{ngram}$. We use ITR$_{ngram}$ to assess the importance of dense retrieval for ITR and its benefits in TableQA.

7.6.2 Results

		WikiSQL				WikiTQ			
#	Models	Dev	Test	Compact	Overflow	Dev	Test	Compact	Overflow
1	TaPEx	88.4	87.7	90.8	58.2	58.7	57.8	62.8	35.3
2	ITR $\rightarrow$ TaPEx	**91.3**	**91.4**	**91.6**	**89.4**	**61.8**	**61.5**	**65.2**	**44.8**
	with N=1	91.0	91.0	**91.6**	85.5	61.4	61.2	**65.2**	43.2
3	ITR$^-_{col}$ $\rightarrow$ TaPEx	89.8	89.5	91.3	72.9	60.9	60.8	64.7	43.4
4	ITR$^-_{row}$ $\rightarrow$ TaPEx	91.1	91.1	91.4	87.8	60.3	59.9	63.8	42.1
5	ITR$^+_{col}$ $\rightarrow$ TaPEx	89.9	89.6	91.2	74.4	60.2	58.8	62.1	43.6
6	ITR$^+_{row}$ $\rightarrow$ TaPEx	90.3	90.5	91.5	80.9	50.9	49.8	53.3	33.7
7	ITR$_{ngram}$ $\rightarrow$ TaPEx	89.5	89.1	91.5	66.6	57.8	57.2	62.4	33.5

Table 7.6 Ablation study of ITR and its variants. Compact and Overflow are subsets of the Test split. ITR is applied both in training and inference. **Bold** denotes the best accuracies for each dataset split.

In Table 7.6 we compare the accuracy of ITR variants on WikiSQL and WikiTQ. In WikiSQL, ITR$^{\{+,-\}}_{row}$ variations (#4 & #6) perform better than the baseline (#1) and the column-wise counterparts (#3 & #5). In WikiTQ we see the opposite: ITR$^{\{+,-\}}_{col}$ variations (#3 & #5) perform better than baseline (#1) and the row-wise counterparts (#4 & #6). However, ITR (#2) demonstrates superior performance across the board by jointly ranking the most relevant rows and columns, which strikes the right balance between the preference for each dataset. Even with N=1, ITR significantly improves the TaPEx baseline. Indeed, N=10 delivers only a slight improvement over N=1, by 0.3-0.4% in Test set depending on the dataset. As such, N=1 is a more efficient solution for applications with limited computational resources. We further discuss varying N values in Appendix C.3.

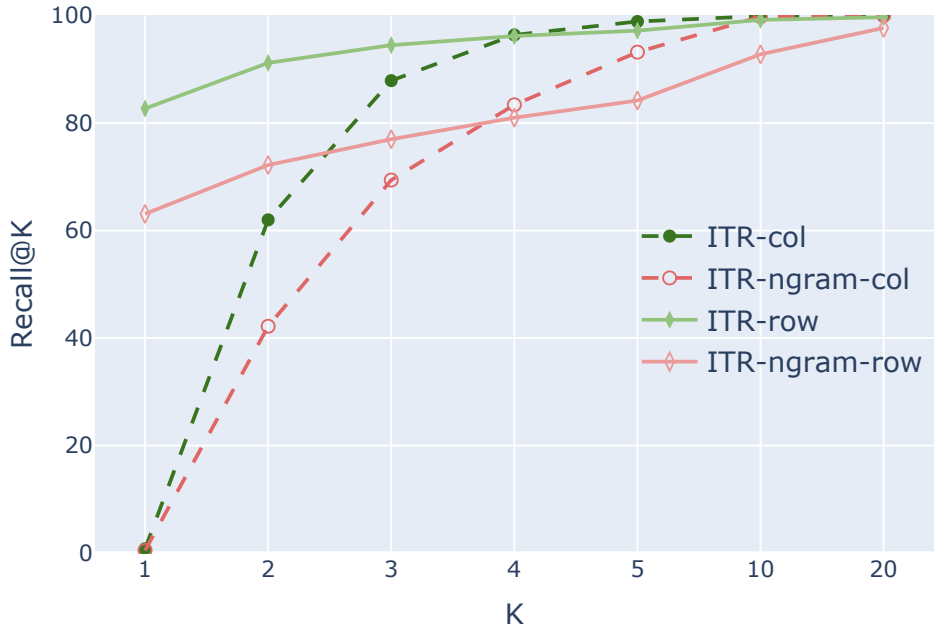


Fig. 7.3 Results on the test set of WikiSQL for ITR and ITR$_{ngram}$ item retrieval. For ITR$_{ngram}$ we set $n \leq 3$, and plot $n = 3$ which shows always better performance.

Semantic versus lexical input selection. In Fig. 7.3, we compare ITR and ITR$_{ngram}$ when performing both row-wise, and column-wise retrieval using Recall@K (whether all the gold cells are in the retrieved table items). Neural-based ITR outperforms ITR$_{ngram}$ for all values of K and both item types. The retrieval performance of ITR converges after $K > 5$ for ITR$_{col}$ and $K > 10$ for ITR$_{row}$. For ITR$_{ngram}$, higher values of K are needed to achieve full Recall@K.

The lower performance of ITR$_{ngram}$ is explained by poor lexical matching of the question with cell values and column names, as compared to the embedding similarities used by neural counterparts. For example, two viable questions that can be answered with the 'Rank' column in Fig. 7.1 are "which mountain peak is the highest?" and the less natural "which is the mountain peak that has the lowest value in the rank?". While the latter matches with the column name, the former is more natural for human to ask and it does not match the column names via n-gram similarity. It is worth mentioning that WikiSQL is more canonicalised and the ITR$_{ngram}$ results in Fig. 7.3 might be positively affected by the nature of WikiSQL. We expect the gap between ITR and ITR$_{ngram}$ to be even larger in challenging datasets where the reference question is more human-like, e.g., WikiTQ. This is indeed evident when these variations are integrated in TableQA. Unsurprisingly, results in Table 7.6 (#7) show that, in contrast to ITR, the ITR$_{ngram}$ variation is able to improve the baseline performance (#1) in WikiSQL, but degrades the performance in WikiTQ.

7.7 Example System Outputs

In this section, we discuss two cases from system outputs. We illustrate the relevance assigned by ITR on the original table rows and columns as a heat-map where the color scale reflects the relevance scores per each column and row with respect to the question (green → yellow → orange → red). To obtain cell scores in their intersection, we sum up their corresponding column-wise and row-wise scores. As a result, more relevant cells are more red.

In Table 7.7 we show a side-by-side comparison of TaPas and TaPEx under the 64 token reduction scenario, and the benefit of applying ITR. TaPEx uses special tokens for encoding the table structure, which make the linearised sequence longer. TaPas instead, encodes the table structure via additional embedding layers (as introduced in Sec. 2.5.2 and Fig. 2.46). In addition, TaPas applies cell truncation to the first token for each cell, which are reconstructed as a post-processing step, and drops rows that exceed the token budget. This allows TaPas to be fed a larger portion of the table in the input, even if the cell information might be lost, e.g., in Table 7.7 Figure A, 'OF-8' is squeezed into only 'OF' removing the distinction between 'Equivalent NATO Rank' across rows. As such, under extreme scenarios TaPas performs better than TaPEx, due to the visibility of a larger portion of the table. ITR proves beneficial and enables TaPEx to view only the relevant information within the token budget (Table 7.7 Figure B, left) to correctly answer the question.

In Table 7.8 we show a comparison of table pruning strategies included in TaPas, such as cell and row truncation, which might cause information loss. We disable those when we apply ITR, and provide TaPas with the full information contained on the question-relevant cells, as determined by ITR. In Table 7.8 Figure C, 'New York' and 'New England Patriots' are both truncated as 'New' by TaPas. This information is crucial for correctly answering the question. Indeed, TaPas fails to locate the right cell after truncation, and predicts a wrong answer. The sub-table created by ITR in Table 7.8 Figure D presents the full information of relevant cells to the model, thus enabling TaPas to make the correct prediction.

7.8 Limitations and Potential Future Work

First, in this work we limit the experimentation to vertical relational web-tables only, following the format of benchmarks used in TableQA, i.e., WikiSQL and WikiTQ. While we believe that ITR can easily be extended to horizontal entity web-tables, e.g., tables from Wikipedia, we do not expect our algorithm to transparently work on other types of tables that we do not consider, e.g., matrix tables from scientific papers and/or spreadsheets [315], where table items can be represented differently. However, this is not a limitation of the

Question: What could a Spanish Coronel be addressed as in the commonwealth military?
Gold Answer: Group Captain.

	Equivalent NATO Rank	Rank in Spanish	Rank in English	Commonwealth equivalent	US Air Force equivalent
0	OF-8	General del Aire	Lieutenant General	Air Marshal	Lieutenant General
1	OF-7	Brigadier General	Major General	Air Vice-Marshal	Major General
2	OF-5	Coronel	Colonel	Group Captain	Colonel
3	OF-4	Teniente Coronel	Lieutenant Colonel	Wing Commander	Lieutenant Colonel
4	OF-3	Mayor	Major	Squadron Leader	Major
5	OF-2	Capitán	Captain	Flight Lieutenant	Captain
6	OF-1	Teniente Primero	First Lieutenant	Flying Officer	First Lieutenant
7	OF-1	Teniente Segundo	Second Lieutenant	Pilot Officer	Second Lieutenant

Figure A: Original Table from WikiSQL with ITR relevance heat-map.

Model	Input [question, table (serialised and tokenised)]	Prediction	Notes
TaPEx	<s> what could a spanish coronel be addressed as in the commonwealth military? col : equivalent nato rank code | rank in spanish | rank in english | commonwealth equivalent | us air force equivalent row 1 : of-8 | general del aire | lieutenant general | air marshal | lieutenant general</s> *(truncated at 64 tokens)*	Air Marshal	TaPEx uses separation tokens interleaving cell values.
TaPas	[CLS] what could a spanish coronel be addressed as in the commonwealth military? [SEP] equivalent rank rank commonwealth us of general lieutenant air lieutenant of brigadier major air major of coronel colonel group colonel of teniente lieutenant wing lieutenant of mayor major squadron major of capitan captain flight captain of teniente first flying first *(62 tokens, with only 1 token in each cell and truncated at 7 rows)*	Group Captain	TaPas uses additional embedding layers to encode table structure. The TaPas tokenizer by default squeezes the number of tokens in each cell to fit the table (e.g., 'OF-8' is squeezed into only OF), during which process information may be lost.

	Rank in Spanish	Rank in English	Commonwealth equivalent
2	Coronel	Colonel	Group Captain
5	Capitán	Captain	Flight Lieutenant

	Rank in Spanish	Rank in English	Commonwealth equivalent	US Air Force equivalent
2	Coronel	Colonel	Group Captain	Colonel
3	Teniente Coronel	Lieutenant Colonel	Wing Commander	Lieutenant Colonel
5	Capitán	Captain	Flight Lieutenant	Captain

Figure B: Largest sub-table obtained for TaPEx (left) and TaPas (right) with ITR relevance heat-map. The largest sub-table for TaPas is bigger than that of TaPEx as the sequence length is calculated based on the tokenization of each TableQA model.

Model	Input [question, sub-table (serialised and tokenised)]	Prediction	Notes
ITR → TaPEx	<s> what could a spanish coronel be addressed as in the commonwealth military? col : rank in spanish | rank in english | commonwealth equivalent row 1 : coronel | colonel | group captain row 2 : capitán | captain | flight lieutenant</s> *(52 tokens without truncation)*	Group Captain	Now, the information being sought is within the length budget, leading to successful answering.
ITR → TaPas	[CLS] what could a spanish coronel be addressed as in the commonwealth military? [SEP] rank in spanish rank in english commonwealth equivalent us air force equivalent coronel colonel group captain colonel teniente coronel lieutenant colonel wing commander lieutenant colonel capitan captain flight lieutenant captain *(53 tokens without truncation)*	Group Captain	Now, the input sub-table is fully presented without harshly squeezing cell tokens.

Table 7.7 Example system outputs with 64 token budget: comparing TaPEx and TaPas with or without ITR. ITR sub-table enables TaPEx to view the relevant information for correctly answering the question.

Question: Which winning team beat the New York Yankees?
Gold Answer: Arizona Diamondbacks.

	Year	Game or event	Date contested	League or governing body	Sport	Winning team	Losing team	Final score
0	2002	2001 World Series, game seven	November 4, 2001	Major League Baseball	Baseball	Arizona Diamondbacks	New York Yankees	3-2
1	2004	Super Bowl XXXVIII	February 1, 2004	National Football League	American football	New England Patriots	Carolina Panthers	32-29
2	2008	Super Bowl XLII	February 3, 2008	National Football League	American football	New York Giants	New England Patriots	17-14
3	2009	Super Bowl XLIII	February 1, 2009	National Football League	American football	Pittsburgh Steelers	Arizona Cardinals	27-23
4	2010	Winter Olympics men's hockey gold-medal game	February 28, 2010	International Olympic Committee	Ice hockey	Canada	United States	3-2 (overtime)
5	2011	NFL Week 15 game	December 19, 2010	National Football League	American football	Philadelphia Eagles	New York Giants	38–31

Figure C: Original Table from WikiSQL with ITR relevance heat-map.

	League or governing body	Winning team	Losing team
0	Major League Baseball	Arizona Diamondbacks	New York Yankees
1	National Football League	New England Patriots	Carolina Panthers
2	National Football League	New York Giants	New England Patriots
4	International Olympic Committee	Canada	United States

Figure D: Largest sub-table obtained for TaPas with ITR relevance heat-map.

Model	Input [question, sub-table (serialised and tokenised)]	Prediction	Notes
TaPas	[CLS] which winning team beat the new york yankees? [SEP] year game date league sport winning losing final 2002 2001 november major baseball arizona new 3 2004 super february national american new carolina 32 2008 super february national american new new 17 2009 super february national american pittsburgh arizona 27 2010 winter february international ice canada united 3 *(52 tokens, with only 1 token in each cell and truncated at 5 rows)*	New York Giants	TaPas tokenizer squeezes cell tokens in order to fit the table, which, however, confuses the model by having only one token "new" for both New York Yankees and New England Patriots. This prevents TaPas from finding the correct answer.
ITR → TaPas	[CLS] which winning team beat the new york yankees? [SEP] league or governing body winning team losing team major league baseball arizona diamondbacks new york yankees national football league new england patriots carolina panthers national football league new york giants new england patriots national football league philadelphia eagles new york giants *(53 tokens without truncation)*	Arizona Diamondbacks	ITR successfully presents most relevant information to TaPas. New York Yankees and New England Patriots are now fully presented, making question answering successful.

Table 7.8 Example system outputs with 64 token budget: TaPas pruning strategies cause information loss, which confuses the model decision. ITR disables such information loss to remediate the previously wrong decision of TaPas.

algorithm itself and adjusting our assumptions to certain scenarios and type of data can be feasible in the future.

Second, ITR selects the relevant table elements by using a question as query. This means that it is specifically designed for tasks that involve both table and text inputs, such as TableQA that we demonstrate in our paper. Additionally, it can be applied to table entailment tasks, like table fact verification, where the goal is to determine if a statement is supported by the data in the table. We leave these extensions as future work. However, ITR cannot be used for tasks where the table is the sole input. An example of such a task is table-to-text generation, where the objective is to generate descriptive text solely from the data within a table. In these cases, ITR's reliance on an accompanying query renders it unsuitable.

Finally, while ITR is beneficial for questions that do not rely on table completeness, its effectiveness is limited when, for example, all table cells are required to be predicted. Consider a question that requires cell counting, and the gold cells satisfying the query can be more than what we can feed a model with, e.g., "how many championship did Player A get?" and Player A has won 500 champions. However, this limitation does not arise from our approach and is rather inherited by existing TableQA models in the literature. Indeed, it can be a potential future direction of our work, which requires model innovation and table transformation that focuses on representing the information in a compact form.

7.9 Summary

In this chapter, we presented ITR for TableQA, an approach to create the most relevant sub-table(s) to efficiently answer a given question via TableQA. ITR is based on a dense retrieval component, which selects relevant rows and columns and combines them into a compact sub-table that satisfies length budget constraints. We combined ITR with different TableQA models from the literature, at training and/or inference time, and showed that ITR indeed captures the most relevant information, which enabled underlying models to perform better overall and become more robust, thus attaining state-of-the-art results on WikiSQL and WikiTQ benchmarks. ITR is flexible, does not depend on the underlying model and can be easily integrated in future model developments to increase their robustness. As future work we can combine ITR with computational operations over different table elements [383] to collapse its information in a more compact format, to benefit also questions that rely on table completeness.

Additionally, the work discussed in this chapter addresses research questions RQ1 and RQ2, details of which will be elaborated in the final chapter of the thesis (Sec. 9.1).

In the next chapter, we move on to explore retrieval models for open-domain TableQA, which is more challenging than close-domain TableQA. We will introduce a novel framework, LI-RAGE: Late Interaction Retrieval Augmented Generation with Explicit Signals for Open-Domain Table Question Answering.

Chapter 8

Late-Interaction Retrieval for Table Question Answering

8.1 Introduction

In the previous chapter we explored retrieval models for closed-domain TableQA. We note that open-domain TableQA, the task of answering questions grounded in external tables, is increasingly attracting attention of both public and commercial research for its value in real-world applications. In this chapter, we introduce a novel method for open-domain TableQA. It is worth noting that the research presented in this chapter was done between Chapter 3 and Chapter 5, which inherited some advanced methods developed in Chapter 3 and inspired the continued work of Chapter 5.

Research open-domain TableQA pipelines are typically implemented with two components: a retriever and a reader. The retriever chooses a small set from the entire pool of table candidates, while the reader generates answers processing each table candidate. State-of-the-art implementations use transformer-based models for both components. In particular, the retriever is built with variants of Dense Passage Retriever [143, DPR], which computes question-table similarity by using single vector representations of the question and the table. Retriever and reader can be trained separately [107] or jointly [236] via Retrieval Augmented Generation loss [167, RAG]. We observe three limitations which we address in this paper.

First, a table can be very large and might contain heterogeneous information across rows/columns; encoding into a fixed size vector risks information loss, which can have an impact in QA quality. One way to alleviate this issue is to replace DPR with a Latent

Interaction (LI) model, which encodes text into token-level representations. In particular, we find ColBERT [146] to be very effective, even if not pre-trained for tables.

Second, RAG uses only an implicit signal to guide the retriever. Recently, Lin and Byrne [183] (our work, introduced in Chapter 3) proposed a loss (Sec. 3.3.3 Eq. 3.6) that can be used to jointly optimises both the retriever and the answer generator, which in this chapter's setting rewards the retriever with table-level signals from the reader model in joint training. In this chapter, for clearer presentation, we adopt the naming convention of referring to this loss as RAGE loss (Retrieval Augmented Generation with Explicit Signals).

Third, we observe empirically that the reader does not always rank answers coming from the gold table at the top. As our reader is a sequence-to-sequence model, we propose a simple modification to the training data: we prepend binary relevance tokens ('yes/no') to the answer itself. The reader learns to generate a first token indicating whether the table is relevant to the question or not. This is similar to the reranking process discussed in Sec. 2.2, yet it uniquely combines reranking with answer generation.

Using these techniques, we build an end-to-end framework, LI-RAGE, (Late Interaction Retrieval Augmented Generation with Explicit Signals) and achieve state-of-the-art results on two benchmarks for open-domain TableQA, NQ-TABLES [107] and E2E-WTQ [235].

The code and data have been released at: https://github.com/amazon-science/robust-tableqa.

8.2 Related Work

This section provides an overview of the related work up to January, 2023, the time of this study.

While open-domain TableQA is yet a relatively unexplored problem, with only a few applications in the past couple of years, there has been extensive work on table retrieval and TableQA separately, as introduced in Sec. 2.5.

In table retrieval, recent advances in machine learning have enabled extracting deep features for tables with Transformers [307], by designing models to parse complex tabular structure [107, 309], or by simply linearising tables with interleaving tokens to preserve its structure [236, 316].

In TableQA, researchers until recently assumed gold tables were given and focused on developing models that understood and answered questions over tables, i.e. the readers. Earlier models generated commands in logical forms (e.g. SQL queries) that were executable over tables [357, 179, 333], while recent state-of-the-art models directly predict the answers from the input question and table by either classification [106, 339, TaPas] or autoregressive generation [200, TaPEx].

Following these advances, in open-domain TableQA the best-performing systems are based on a retriever-reader pipeline [107, 236]. Herzig et al. [107, DTR] leverages TaPas [106] to both initialise a DPR-like retriever and the reader. T-RAG [236] uses DPR as retriever of rows/columns by decomposing the table and generates the answer via a sequence-to-sequence reader [166], applying the RAG loss to refine the retriever with implicit signals during end-to-end TableQA fine-tuning. Unlike DTR and T-RAG, CLTR [235] employs only retrieval of rows and columns and obtains the answer cell by intersecting the top-scored ones. In this work we focus mainly on the retriever, and unlike previous work that relies on single vector embeddings, we leverage late interaction retrievers [146] to achieve a finer-grained interaction between questions and tables. In contrast to T-RAG and CLTR, we do not need to decompose the table into rows and columns, but retrieve a whole table from the corpus, ensuring that the reader is given all the relevant information. In addition, we explore different techniques for *explicitly* refining the retriever during end-to-end TableQA, achieving superior performance.

8.3 Methodology

Given a question q, the tasks are to find the *gold* table t^* from a table corpus $\mathscr{T}$, i.e. table retrieval (Sec. 8.3.1), and to derive the answer denotations $\mathscr{S}$ (1 or more cells from the table), i.e. question answering over the retrieved tables (Sec. 8.3.2). We assume that labeled datasets consisting of triples $\{(q, \mathscr{S}, t^*)\}$ are available to us. We flatten the tables into sequences with interleaving special tokens that encode its structure (see Appendix D.1).

8.3.1 Table Retrieval

In order to exploit question-table similarity at a finer-grained level than when using DPR models, we leverage LI models to encode and retrieve tables for a question. We use ColBERT, which consists of a question encoder $\mathscr{F}_q$ and a table encoder $\mathscr{F}_t$, to encode questions and tables at the *token level*:

$$\mathbf{Q} = \mathscr{F}_q(q) \in \mathbf{R}^{l_q \times d}; \mathbf{T} = \mathscr{F}_t(t) \in \mathbf{R}^{l_t \times d}, \tag{8.1}$$

where l_q and l_t are input token lengths of q and t. The relevance score accounts for the interactions between all question and table token embeddings:

$$r(q,t) = \sum_{i=1}^{l_q} \max_{j=1}^{l_t} \mathbf{Q}_i \mathbf{T}_j^\top. \tag{8.2}$$

LI models extract multi-dimensional question/table embeddings and token-level similarity, as opposed to finding the similarity of single embeddings for the whole question/table in DPR, thus capturing a finer-grained interaction between them.

To train the model we exploit the gold (positive) table t^* for each question q, i.e., explicitly considering the table-level ground truth. We use in-batch negative sampling for training, per Karpukhin et al. [143]. All documents in a training batch other than t^* are considered negative for q, and denoted as $\mathcal{N}(q)$. We train with the contrastive loss $\mathscr{L}_{CL}$:

$$-\sum_{(q,t^*)} \log \frac{\exp\left(r(q,t^*)\right)}{\exp\left(r(q,t^*)\right)+\sum_{z\in\mathcal{N}(q)} \exp\left(r(q,z)\right)}. \tag{8.3}$$

To this end, for each q, the retriever outputs K top-scoring tables $\{t_k\}_{k=1}^{K}$. Finally, following RAG and RA-VQA, we obtain their (approximate[1]) conditional probability $p_\theta(\cdot|q)$ with the retriever parameters θ:

$$p_\theta(t_k|q) = \frac{\exp(r(q,t_k))}{\sum_{j=1}^{K}\exp(r(q,t_j))}. \tag{8.4}$$

8.3.2 Retrieval-based TableQA

For the TableQA task we make use of a sequence-to-sequence Transformer-based model that directly produces an answer for a given question and table. The TableQA model p_ϕ takes as input a sequence composed of the question q and each of the retrieved tables t_k, as described in Sec. 8.3.1, and generates an answer y_k for each input table t_k:

$$y_k = \underset{y}{\operatorname{argmax}}\, p_\phi(y|q,t_k). \tag{8.5}$$

Finally, the model returns the answer associated with the highest probability/confidence:

$$\widehat{y},\widehat{t} = \underset{y,t_k}{\operatorname{argmax}}\, p_\phi(y|q,t_k). \tag{8.6}$$

8.3.3 Joint Training of Retrieval and TableQA

We train both modules jointly using a compositional loss that is borrowed from RA-VQA (Sec. 3.3.3) [183], which considers signals from table relevance and answer prediction, as

[1] because we sum over the top-K tables instead of all tables, assuming the probabilities of the rest are small and irrelevant.

follows:

$$-\sum_{(q,\mathscr{S})}\Big(\sum_{k=1}^{K}\log p_\phi(s_k^*|q,t_k)+\sum_{k\in\mathscr{P}^+(q,\mathscr{S})}\log p_\theta(t_k|q)\Big), \tag{8.7}$$

where s_k^* is a concatenation of all comma-separated answers in $\mathscr{S}$ and $\mathscr{P}^+(q,\mathscr{S})=\{k: y_k = s_k^* \wedge t_k = t^*\}$ is a subset of the retrieved K tables, which contains those tables that satisfy (1) being a gold table relevant to answering the question; (2) the answer generator successfully produces the correct answer from that table. The core idea is to leverage the signal from model prediction to decide which tables are beneficial to producing the correct answer. Their scores are dynamically adjusted during training, which tailors the retriever to better serve the answer generation.

8.3.4 Learned Table Relevance

The answer generator is trained to produce s_k^* for each input (q,t_k) pair. Ideally, we would assume that the answer generated from the gold table t^* is also associated with the highest probability from the answer generator. However, it might happen that an answer derived from a non-gold retrieved table may achieve higher confidence than the answer derived from a gold retrieved table. We propose a simple yet effective approach to improve this process: we add a *binary relevance token* preceding s_k^* as 'yes' if $t_k = t^*$, 'no' otherwise. This design aims at guiding the model to prioritise reliable answer sources at training time. At generation time, if the leading generation of a (q,t_k) pair is 'yes', we consider t_k to be a more reliable answer source and prioritise it over other input tables—that generate 'no' instead—when selecting the final prediction. We rely on the confidence scores if the leading token of all the candidates is 'no'.

The use of binary relevance tokens mirrors the re-ranking process discussed in Sec. 2.2, and we effectively integrate re-ranking into the generation pipeline. For this reason, we retain negative items (where the binary relevance token is 'no') during training, as they are essential for training the system to identify irrelevant passages.

8.4 Experimental Setup

8.4.1 Datasets and metrics

We evaluate our system on two benchmarks, i.e. NQ-TABLES [107] and E2E-WTQ [235]. Their statistics are provided in Table 8.1.

Dataset	Train	Dev	Test	#Tables
NQ-TABLES	9,594	1,068	966	169,898
E2E-WTQ	851	124	241	2,108

Table 8.1 Dataset statistics of NQ-TABLES [107] and E2E-WTQ [235].

NQ-TABLES contains generally hard questions extracted from the NaturalQuestions [155] dataset, comprising the questions that can be answered from tables rather than plain text. For this benchmark, we evaluate the models using: Token F1, i.e. token-wise F1 score; and exact match (EM) or accuracy, i.e. whether predictions match the annotations.

E2E-WTQ contains look-up questions that require cell selection operation and is a subset of WikiTableQuestions [237]. In E2E-WTQ train/valid/test splits are the same as in WikiTableQuestions, with questions limited to those that do not require aggregations across multiple table cells. We evaluate models via accuracy[2].

In addition, we report Recall@K for the retrieval performance in both, which measures whether the gold table is among the top-K retrieved tables.[3]

8.4.2 System configurations

For the table retrieval component, we conduct contrastive experiments using both DPR and LI. We first fine-tune the official pre-trained DPR or ColBERTv2 model on each dataset before using them in the joint retriever-reader training. We do not train the TableQA model from scratch, instead we warm-start the training with TaPEx, a state-of-the-art pre-trained model for tabular data understanding based on BART [166] (see Sec. 2.5, Fig. 2.47 for details). Since the E2E-WTQ is very small and not enough for learning a robust TableQA model, we additionally fine-tune TaPEx on its superset, i.e. WikiTableQuestions. Note that no test samples are leaked due to this as the dataset splits of E2E-WTQ are the same as WikiTableQuestions. We select the best checkpoints based on the validation performance. We set K=5 since it shows the best balance between performance and latency by both RAG and our work in Chapter 3. Training details, computational cost and software solution are provided in Appendix D.3.

[2]Also named as Hit@1 in Pan et al. [235, 236]

[3]We do not report metrics such as P@K, N@K, MAP used by T-RAG and CLTR, which decompose tables, being incompatible with our setting (see Appendix D.2).

8.4.3 Comparison Systems

We compare with models from the literature: **DTR**, **CLTR**, **T-RAG** (see Sec. 8.2), and **BM25**—sparse retrieval baseline introduced in Sec. 2.2.1. Moreover, we build the following model variants:

(1) **LI-RAGE**: our main system that leverages ColBERT as retriever, TaPEx as answer generator, RAGE loss for joint training and the binary relevance token in output. We also ablate the system showing the effectiveness of each feature. When disabling joint training, i.e., for ablating the model, the retriever is not updated.

(2) **DPR-RAGE**: similar to LI-RAGE, except the retriever is a DPR model.

(3) **RAG**: we train the RAG [167] in TableQA data, initialising the retriever and answer generator with our fine-tuned DPR and TaPEx, respectively. Different from DPR-RAGE, RAG does not produce the binary relevance token and updates the retriever only with the RAG loss, which is an implicit signal from the reader.

8.5 Results and Discussions

8.5.1 Main Results

Models	NQ-TABLES			E2E-WTQ	
	Token F1	EM	Recall@K	Accuracy	Recall@K
DTR+hn [107]	47.70	37.69	81.13@10	-	-
CLTR [235]	-	-	-	46.75	-
T-RAG [236]	50.92	43.06	85.40@10	50.65	-
RAG	39.67	38.33	69.16@5	38.05	61.29@5
DPR-RAGE	49.68	43.02	84.35@5	48.79	59.68@5
LI-RAGE	**54.17**	**46.15**	**87.90**@5	**62.10**	**81.85**@5
(w/o joint training)	53.53	45.52	85.21@5	59.27	81.45@5
(w/o relevance tokens)	50.56	42.53	86.90@5	53.69	81.75@5
(w/o joint training & relevance tokens)	49.83	42.19	85.21@5	50.16	81.45@5

Table 8.2 End-to-end TableQA performance on NQ-TABLES and E2E-WTQ. Best performances are in **bold**.

As shown in Table 8.2, LI-RAGE achieves the best performance across the board on both datasets, with more than 3 points improvements in Token F1 and EM in NQ-TABLES, and 11.45 points in E2E-WTQ with respect to previously best reported results in the literature. We attribute these results to the high performance of the LI retriever. On NQ-TABLES it

Models	NQ-TABLES				E2E-WTQ			
	K=1	K=5	K=10	K=50	K=1	K=5	K=10	K=50
BM25	17.62	35.97	43.80	61.00	58.09	74.27	79.67	87.55
DPR-RAGE	58.29	84.35	90.72	97.08	33.61	59.68	66.80	88.38
(w/o joint training)	53.07	84.25	90.62	**97.81**	32.78	58.47	66.39	88.38
LI-RAGE	**59.12**	**87.90**	**92.81**	97.60	**68.46**	**81.85**	**85.89**	93.36
(w/o joint training)	53.75	85.21	90.10	97.71	66.13	81.45	84.27	**93.55**

Table 8.3 Retrieval performance on NQ-TABLES and E2E-WTQ. Best performances are in **bold**.

obtains the best recall rate (87.90%) when only 5 tables are retrieved, as opposed to the previous models that achieve a lower recall rate with $K = 10$ tables, and also performs better when compared with RAG and DPR-RAGE, by a large margin.

Effects of Joint Training

Similar to the observation in Chapter 3, joint training with RAGE improves over the frozen system on both retrieval and TableQA performance. As shown in Table 8.2, joint training improves the end-to-end TableQA performance on both datasets by $\sim$0.6% and $\sim$2.83%, respectively, and shows a superior retrieval ability especially on NQ-TABLES (85.21 to 87.90).

Effects of Binary Relevance Tokens

As shown in Table 8.2, removing the binary relevance tokens greatly reduces system performance, by around 3.6% Token F1 and EM in NQ-TABLES and 8.4% in E2E-WTQ accuracy.

Effects of LI

We report the retrieval performance in Table 8.3. LI-RAGE achieves the highest recall, outperforming BM25 in both datasets, and DPR by $\sim$3% on NQ-TABLES and by over 20-30% Recall@5/1 on E2E-WTQ. The large margin on E2E-WTQ is because it contains generally long tables with diverse information, and LI models prove beneficial in learning richer table representations.

8.5.2 Remarks on Design Rationale

We tailor our solution for TableQA, with the specific design of two main components, i.e., adding a relevance token and modifying the RAGE loss.

Relevance token

In Open-domain QA, open-ended questions can have multiple correct answers and may be answered by different passages. Consequently, increasing the number of retrieved passages (K) often enhances retrieval performance by broadening the search coverage. However, this is not the case for tables. In open-domain TableQA, questions typically have only one relevant table, and most questions focus on a specific cell within that table. Our preliminary experiments demonstrated that increasing K reduced performance when $K > 5$, as presenting more tables to the answer generator increases confusion and the likelihood of errors, due to overconfidence in incorrectly retrieved tables. When utilising relevance tokens as per our design, increasing K does not negatively affect performance because irrelevant tables are excluded.

Additionally, we explored alternative strategies that use retrieval scores to assess document reliability. The first strategy involves predicting the final answer from the table with the highest retrieval score. This approach achieved 41.04 EM score on NQ-TABLES, which is lower than our ablated LI-RAGE configuration *without joint training & relevance tokens*, which attained an EM score of 42.19 (see Table 8.2).

A second strategy weights predictions from different tables by their corresponding retrieval scores, i.e., multiplying the retrieval score (from the retriever) by the answer confidence (from the answer generator) when using $K = 5$. This approach also performed worse than our ablated LI-RAGE configuration *without joint training & relevance tokens*, achieving EM scores of 40.91 and 42.19 on NQ-TABLES, respectively.

In summary, relevance tokens outperform both document retrieval scores and the combination of retriever and reader scores.

RAGE loss

We modify the original RAGE loss in Chapter 3 to adapt it to the domain of tables. In particular, we dropped the third term in the equation, which penalises documents if they do not contain gold answers and also do not contribute to successful question-answering. Enabling this term in the loss penalises $K-1$ documents in most cases, which leads to collapsed performance of the retriever in joint training for TableQA. This is motivated by the same fact that gold tables are relatively sparse in TableQA and penalising incorrect

documents leads to instability of training and quick retriever overfitting. Disabling this term instead, softens the RAGE loss by only rewarding "good" tables and distinguishing good tables from bad ones, which improved the performance by around 1% EM on NQ-TABLES.

8.5.3 Computational Cost

Models	Training Speed (iter/sec)	Training Batch Size	Training Time (mins)	Inference Speed (sec/iter)	Inference Batch Size
DPR	1.10	8	60 (NQ)/ 10 (WTQ)	-	-
LI	1.75	6	60 (NQ)/ 10 (WTQ)	-	-
DPR-RAGE	2.1	1	300 (NQ)/ 35 (WTQ)	1.22	4
LI-RAGE	0.74	1	450 (NQ)/ 50 (WTQ)	1.40	4

Table 8.4 Computational cost for DPR/LI retriever models and LI-RAGE and DPR-RAGE.

In Table 8.4 we report computational cost of the proposed models. It is clear that time spent on the training of LI is not significantly increased compared to DPR training. This is because both models use contrastive learning in training. But we note that the index building time of LI is around 5 mins while that of DPR only takes 40 seconds.

In terms of joint training, the end-to-end training time of LI-RAGE is longer. This is due to (1) slightly slower dynamic retrieval during end-to-end training; (2) refining the retriever via larger multi-dimensional embeddings in comparison to one-dimensional embeddings used in DPR-RAGE. However, the inference speed is not affected much (from 1.22 sec/iteration to 1.40). This suggests that when deployed as real applications, LI-RAGE does not bring significant increase in computation.

8.6 Limitations and Potential Future Work

Our proposed system was tested on two open-domain TableQA datasets, with one of them (E2E-WTQ) being relatively small compared to the other. Also, the current open-domain TableQA datasets are limited to simple questions. They do not cover more complicated questions that involve multiple cells and complex table operations, such as SUM, MAX, MIN, and SUBTRACT in some questions of WikiSQL and WikiTableQuestion. Therefore, the effectiveness of our system can be further evaluated on more complicated datasets of larger scale in the future. However, our system is potentially well-suited to handle even more complex questions, as its design enables the retrieval of entire rows or columns when necessary, such as when calculating the average value across the cells in a column. Another

limitation lies in the token length limit of modern Transformer models. The best-achieving models as of date typically accept up to 1024 tokens (e.g. BART, the base model of TaPEx). This limitation becomes more obvious when tables grow longer and the information being sought go beyond the limit. We believe that, with better approaches addressing this limitation, our system can achieve better performance. The solution can be either applying sampling strategies to pick the rows and columns that are most relevant to answering the question (as did in Chapter 7), or increasing the capacity of future Transformer models, which recent LLMs (introduced in Sec. 2.1.3) have extended to over ten thousand tokens.

8.7 Summary

In this chapter, we introduce a novel open-domain TableQA framework, LI-RAGE, that leverages late interaction retrievers to enable finer-grained interaction between questions and tables. Additionally, LI-RAGE incorporates the RAGE loss and binary relevance tokens which enable significant improvements over the state-of-the-art in two challenging TableQA tasks.

Additionally, the work discussed in this chapter addresses research questions RQ1, RQ2, and RQ3, details of which will be elaborated in the final chapter of the thesis (Sec. 9.1).

In the next chapter, we summarise all the research presented in this thesis, and discuss potential follow-up work in the future.

Chapter 9

Conclusion

In this chapter, we provide a comprehensive summary of the work presented within this thesis.

In Chapter 1, we presented an overview of the evolution of LLM/LMM/RAG within both industrial and academic spheres. This overview serves to show the necessity and efficacy of developing multi-modal retrieval augmented systems, which is the core of this research.

Chapter 2 delved into an extensive literature review covering recent advancements in LLM and LMM (Sec.2.1), Information Retrieval (Sec.2.2), and Retrieval Augmented Generation (Sec.2.3). Next, we introduced Visual Question Answering (Sec.2.4) and Table Question Answering (Sec. 2.5) as the primary tasks investigated in this thesis, along with an examination of recent models/systems developed for these tasks.

Commencing from Chapter 3, we demonstrated the models and systems proposed to address multi-modal QA challenges:

Chapter 3 presented RA-VQA, an RAG framework tailored for KB-VQA. It adopts a joint training approach for the retriever and generator, achieving robust KB-VQA performance.

In Chapter 4, we scrutinised data imbalance issues inherent in the FVQA dataset and introduced a new research dataset, FVQA 2.0. This semi-automatically annotated dataset serves to substantially enhance the robustness of KB-VQA systems.

Chapter 5 showcased FLMR, a strong multi-modal retriever bolstered by late-interaction models. The RA-VQA-v2 framework, integrating FLMR, attains state-of-the-art retrieval performance across the board.

Chapter 6 delved into an empirical exploration of the scalability of FLMR. The PreFLMR model, a scaled-up iteration of the original FLMR, encompasses enhancements in training time, dataset size, model parameters, and more.

Chapters 7 and 8 pivoted the focus towards retrieval models in TableQA.

In Chapter 7, we unveiled a novel ITR model capable of substantially enhancing the robustness of TableQA systems, thereby achieving state-of-the-art performance on popular TableQA datasets. Furthermore, we highlighted the adaptability of ITR across diverse TableQA models for performance enhancement.

Chapter 8 introduced the LI-RAGE framework tailored for open-domain TableQA. This framework combines several existing methodologies to yield state-of-the-art performance across popular open-domain TableQA tasks.

To summarise, this thesis traverses the landscape of multi-modal retrieval augmented systems, elucidating their evolution, necessity, and efficacy. Through a meticulous exploration of literature, we establish a comprehensive understanding of the prevailing methodologies and challenges within the realm of multi-modal retrieval and question answering.

Moreover, our contributions extend beyond theoretical elucidation, manifesting in the development of novel frameworks and datasets aimed at addressing prominent challenges in multi-modal question answering. From RA-VQA to LI-RAGE, each framework represents a significant stride towards enhancing the robustness and performance of multi-modal systems across diverse domains.

As we conclude this thesis, it is evident that the journey towards effective multi-modal retrieval augmented systems is ongoing, marked by continual innovation and refinement. The frameworks and insights presented herein serve as a foundation upon which future advancements can be built, paving the way for more sophisticated systems capable of meeting the evolving demands of information retrieval and question answering in multi-modal contexts.

9.1 Key Findings

Now, we can revisit the research questions posed at the beginning of the thesis and discuss how the findings presented support them.

RQ1: Can retrieval methods be utilised to enhance multi-modal systems' ability to acquire knowledge from external sources when answering domain-specific or challenging questions?

The answer is yes. Throughout Chapters 3 to 6, we introduced the RA-VQA and RA-VQA-v2 frameworks to demonstrate that the performance on KB-VQA, which involves challenging multi-modal questions, can be significantly enhanced through retrieval-augmented methods. Additionally, in Chapter 8, we introduced LI-RAGE, which further corroborates the effectiveness of retrieval methods in enhancing the capabilities of systems on multi-modal, challenging TableQA questions. Through both quantitative analysis and qualitative studies,

we have conclusively demonstrated that retrieval methods can effectively acquire knowledge from external sources and the integration of retrieval methods is essential for successful multi-modal QA.

RQ2: Can retrieval methods help multi-modal systems focus more effectively on the current task by filtering out irrelevant and redundant information?

The answer is yes. In Chapter 7, we utilised the ITR framework to identify and remove irrelevant or redundant elements from tables before they are processed by the TableQA model. This approach not only enhanced the performance of TableQA on ordinary tables, which already fit within the model's capacity, but also provided significant improvements on overflow tables filled with excessive redundant information. Furthermore, across all chapters, we observed substantial improvements in downstream QA performance metrics, such as accuracy, following enhancements in retrieval performance metrics like recall. These findings indicate that effective retrieval performance enhances the relevance of retrieved documents and minimises irrelevant content in the RAG framework, ultimately leading to superior QA outcomes.

RQ3: What strategies can be employed to effectively integrate retrieval methods with multi-modal systems?

In Chapters 3 and 8, we investigated a joint training scheme for multi-modal RAG systems. We demonstrated that by leveraging signals from answer generation, the retriever can be jointly optimised during training. This joint training approach customises the retriever specifically for the downstream task, as it utilises direct feedback on the success of the QA process. In two different applications, both using pseudo relevance labels in KB-VQA and ground-truth relevance labels in TableQA, we observed significant enhancements in both retrieval and QA performance. These results suggest that our proposed strategy offers strong generalisability across various tasks and application scenarios.

Conclusion: From these key findings, we have successfully addressed the central research questions and can conclude that multi-modal QA systems can be significantly enhanced through the integration of information retrieval methods.

9.2 Future Work

Looking ahead, several promising avenues beckon for future research to build on the multi-modal retrieval augmented systems presented in this thesis.

One such direction involves bringing together techniques recently developed for LLMs and LMMs. For example, techniques such as prompt engineering, in-context learning, and Chain-of-Thought prompting (introduced in Sec. 2.1.3) hold significant promise for

enhancing the reasoning and analytical capabilities of the underlying LLMs and LMMs when integrated into frameworks like RA-VQA (Chapter 3), RA-VQA-v2 (Chapter 5), LI-RAGE (Chapter 8), and ITR (Chapter 7).

Furthermore, while the RAG pipelines demonstrated in this thesis offer a straightforward two-stage solution, there exists potential for enhancing system performance by integrating more retrieval/reranking stages. Incorporating reranking models (either specialised models or pre-trained LLM/LMMs) is promising to improve the relevance of top-K documents utilised for response generation.

Another promising avenue is leveraging LLMs/LMMs in retrieval processes, albeit with a careful consideration of the trade-off between computational costs and retrieval accuracy. Balancing these factors will be crucial in optimising the efficacy of such approaches. One promising direction is to incorporate the retrieval task in the pre-training of LLMs or LMMs. The RAG process can be speeded up significantly if the LLM can be multi-tasked to generate embeddings for retrieval in addition to generating responses. Initially, the query is encoded by the LLM, which generates query embeddings for retrieval. The internal hidden states of the query can be reused once relevant documents are retrieved and appended to the query, as demonstrated by Muennighoff et al. [225]. However, existing research focuses on generating one-dimensional query embeddings. Extending this to generate multi-modal query embeddings based on multi-dimensional, late-interaction embeddings, as did by FLMR (Chapter 5) and PreFLMR (Chapter 6), may lead to superior performance improvements.

9.3 Ethical Considerations

Our research mainly concerns retrieval models and RAG systems. In work concerning retrieval models, we acknowledge the potential for the retrieved documents to include inappropriate information if the document database lacks adequate filtering. Consequently, extra care must be taken to ensure the sanitisation of the document database, particularly when employing these models in applications involving direct interaction with real users.

In terms of the proposed RAG systems, there is a possibility that the trained QA systems might generate inappropriate or biased content as a result of the training data biases during LLM and LMM pre-training and fine-tuning. Therefore, it is advised to conduct an ethical review prior to deploying these system in live service.

Regarding the FVQA 2.0 dataset proposed in Chapter 4, the dataset was created semi-automatically from the FVQA dataset and ConceptNet, a crowd sourced common sense knowledge graph. Though we have included human annotators in the loop to remove sexual, offensive, and other inappropriate data samples that were automatically generated

(we removed ∼200 inappropriate knowledge graph triplets during annotation), we recognise that the dataset may still contain a small number of inappropriate samples. Any developers who replicate the semi-automatic methodology to extend the datasets should include a similar review step in their manual work flow. We also recognise that the systems trained on this dataset may convey such inappropriate information to users in real-life applications. Therefore, it is crucial to exercise caution when utilising this dataset in applications that engage directly with real users.

9.4 Summary

In this chapter, we have encapsulated the contributions of the thesis and offered glimpses into potential avenues for future research.

As we draw this thesis to a close, I extend my sincere gratitude for your time and engagement in reading this work. May the insights and frameworks presented herein serve as catalysts for further exploration and innovation in the field of multi-modal retrieval augmented systems.

References

[1] Aniket Agarwal, Ayush Mangal, et al. Visual relationship detection using scene graphs: A survey. *arXiv preprint arXiv:2005.08045*, 2020.

[2] Eneko Agirre, Xabier Arregi, and Arantxa Otegi. Document expansion based on wordnet for robust ir. In *Coling 2010: Posters*, pages 9–17, 2010.

[3] Anthropic AI. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 2024. URL https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf.

[4] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[5] Ibrahim Alabdulmohsin, Behnam Neyshabur, and Xiaohua Zhai. Revisiting neural scaling laws in language and vision. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/8c22e5e9181987027 65ecff4b20d0a90-Paper-Conference.pdf.

[6] Marwah Alaofi, Luke Gallagher, Mark Sanderson, Falk Scholer, and Paul Thomas. Can generative llms create query variants for test collections? an exploratory study. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1869–1873, 2023.

[7] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 23716–23736. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/960a172bc7fbf0177ccccbb411a7d800-Paper-Conference.pdf.

[8] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6077–6086, 2018.

[9] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sünderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.

[10] Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, Sebastian Ruder, Yi Tay, Kefan Xiao, Yuanzhong Xu, Yujing Zhang, Gustavo Hernandez Abrego, Junwhan Ahn, Jacob Austin, Paul Barham, Jan Botha, James Bradbury, Siddhartha Brahma, Kevin Brooks, Michele Catasta, Yong Cheng, Colin Cherry, Christopher A. Choquette-Choo, Aakanksha Chowdhery, Clément Crepy, Shachi Dave, Mostafa Dehghani, Sunipa Dev, Jacob Devlin, Mark Díaz, Nan Du, Ethan Dyer, Vlad Feinberg, Fangxiaoyu Feng, Vlad Fienber, Markus Freitag, Xavier Garcia, Sebastian Gehrmann, Lucas Gonzalez, Guy Gur-Ari, Steven Hand, Hadi Hashemi, Le Hou, Joshua Howland, Andrea Hu, Jeffrey Hui, Jeremy Hurwitz, Michael Isard, Abe Ittycheriah, Matthew Jagielski, Wenhao Jia, Kathleen Kenealy, Maxim Krikun, Sneha Kudugunta, Chang Lan, Katherine Lee, Benjamin Lee, Eric Li, Music Li, Wei Li, YaGuang Li, Jian Li, Hyeontaek Lim, Hanzhao Lin, Zhongtao Liu, Frederick Liu, Marcello Maggioni, Aroma Mahendru, Joshua Maynez, Vedant Misra, Maysam Moussalem, Zachary Nado, John Nham, Eric Ni, Andrew Nystrom, Alicia Parrish, Marie Pellat, Martin Polacek, Alex Polozov, Reiner Pope, Siyuan Qiao, Emily Reif, Bryan Richter, Parker Riley, Alex Castro Ros, Aurko Roy, Brennan Saeta, Rajkumar Samuel, Renee Shelby, Ambrose Slone, Daniel Smilkov, David R. So, Daniel Sohn, Simon Tokumine, Dasha Valter, Vijay Vasudevan, Kiran Vodrahalli, Xuezhi Wang, Pidong Wang, Zirui Wang, Tao Wang, John Wieting, Yuhuai Wu, Kelvin Xu, Yunhan Xu, Linting Xue, Pengcheng Yin, Jiahui Yu, Qiao Zhang, Steven Zheng, Ce Zheng, Weikang Zhou, Denny Zhou, Slav Petrov, and Yonghui Wu. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*, 2023.

[11] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[12] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8.

[13] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer, 2007.

[14] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[15] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[16] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. Ms marco: A human generated machine reading comprehension dataset. (arXiv:1611.09268), October 2018. doi: 10.48550/arXiv.1611.09268. URL http://arxiv.org/abs/1611.09268. arXiv:1611.09268 [cs].

[17] Hangbo Bao, Wenhui Wang, Li Dong, Qiang Liu, Owais Khan Mohammed, Kriti Aggarwal, Subhojit Som, Songhao Piao, and Furu Wei. Vlmo: Unified vision-language pre-training with mixture-of-modality-experts. *Advances in Neural Information Processing Systems*, 35:32897–32912, 2022.

[18] Arne Bewersdorff, Christian Hartmann, Marie Hornberger, Kathrin Seßler, Maria Bannert, Enkelejda Kasneci, Gjergji Kasneci, Xiaoming Zhai, and Claudia Nerdel. Taking the next step with generative artificial intelligence: The transformative role of multimodal large language models in science education. *arXiv preprint arXiv:2401.00832*, 2024.

[19] Nitin Bhatia et al. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*, 2010.

[20] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[21] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International conference on machine learning*, pages 2206–2240. PMLR, 2022.

[22] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 446–461, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10599-4.

[23] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

[24] Emanuele Bugliarello, Fangyu Liu, Jonas Pfeiffer, Siva Reddy, Desmond Elliott, Edoardo Maria Ponti, and Ivan Vulić. Iglue: A benchmark for transfer learning

across modalities, tasks, and languages. In *Proceedings of the 39th International Conference on Machine Learning*, page 2370–2392. PMLR, June 2022. URL https://proceedings.mlr.press/v162/bugliarello22a.html.

[25] Jannis Bulian, Christian Buck, Wojciech Gajewski, Benjamin Börschinger, and Tal Schuster. Tomayto, tomahto. beyond token-level answer equivalence for question answering evaluation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 291–305, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.20. URL https://aclanthology.org/2022.emnlp-main.20.

[26] Andrea Burns, Krishna Srinivasan, Joshua Ainslie, Geoff Brown, Bryan A. Plummer, Kate Saenko, Jianmo Ni, and Mandy Guo. Wikiweb2m: A page-level multimodal wikipedia dataset, May 2023. URL https://arxiv.org/abs/2305.05432v1.

[27] Minwoo Byeon, Beomhee Park, Haecheon Kim, Sungjun Lee, Woonhyuk Baek, and Saehoon Kim. Coyo-700m: Image-text pair dataset. https://github.com/kakaobrain/coyo-dataset, 2022.

[28] Guihong Cao, Jian-Yun Nie, Jianfeng Gao, and Stephen Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 243–250, 2008.

[29] Banghao Chen, Zhaofeng Zhang, Nicolas Langrené, and Shengxin Zhu. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv preprint arXiv:2310.14735*, 2023.

[30] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1171. URL https://aclanthology.org/P17-1171.

[31] Guiming Hardy Chen, Shunian Chen, Ruifei Zhang, Junying Chen, Xiangbo Wu, Zhiyi Zhang, Zhihong Chen, Jianquan Li, Xiang Wan, and Benyou Wang. Allava: Harnessing gpt4v-synthesized data for a lite vision-language model. *arXiv preprint arXiv:2402.11684*, 2024.

[32] Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 2024.

[33] Jinghong Chen, Weizhe Lin, and Bill Byrne. Schema-guided semantic accuracy: Faithfulness in task-oriented dialogue response generation. *arXiv preprint arXiv:2301.12568*, 2023.

[34] Jinghong Chen, Weizhe Lin, Jingbiao Mei, and Bill Byrne. Control-DAG: Constrained decoding for non-autoregressive directed acyclic t5 using weighted finite state automata. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational*

Linguistics: Human Language Technologies (Volume 2: Short Papers), pages 508–518, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-short.42.

[35] Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechu Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

[36] Wenhu Chen, Hexiang Hu, Xi Chen, Pat Verga, and William Cohen. MuRAG: Multimodal retrieval-augmented generator for open question answering over images and text. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5558–5570, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.emnlp-main.375.

[37] Wenhu Chen, Hexiang Hu, Chitwan Saharia, and William W. Cohen. Re-imagen: Retrieval-augmented text-to-image generator. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XSEBx0iSjFQ.

[38] Xi Chen, Xiao Wang, Soravit Changpinyo, A. J. Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model. (arXiv:2209.06794), Sep 2022. doi: 10.48550/arXiv.2209.06794. URL http://arxiv.org/abs/2209.06794. arXiv:2209.06794 [cs].

[39] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

[40] Xi Chen, Josip Djolonga, Piotr Padlewski, Basil Mustafa, Soravit Changpinyo, Jialin Wu, Carlos Riquelme Ruiz, Sebastian Goodman, Xiao Wang, Yi Tay, et al. Pali-x: On scaling up a multilingual vision and language model. *arXiv preprint arXiv:2305.18565*, 2023.

[41] Xinyun Chen, Petros Maniatis, Rishabh Singh, Charles Sutton, Hanjun Dai, Max Lin, and Denny Zhou. Spreadsheetcoder: Formula prediction from semi-structured context. In *International Conference on Machine Learning*, 2021.

[42] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural*

Language Processing, pages 14948–14968, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.925. URL https://aclanthology.org/2023.emnlp-main.925.

[43] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? *arXiv preprint arXiv:2302.11713*, 2023.

[44] Yang Chen, Hexiang Hu, Yi Luan, Haitian Sun, Soravit Changpinyo, Alan Ritter, and Ming-Wei Chang. Can pre-trained vision and language models answer visual information-seeking questions? In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14948–14968, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.925. URL https://aclanthology.org/2023.emnlp-main.925.

[45] Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer, 2020.

[46] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Zhong Muyan, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv preprint arXiv:2312.14238*, 2023.

[47] Zhuo Chen, Jiaoyan Chen, Yuxia Geng, Jeff Z Pan, Zonggang Yuan, and Huajun Chen. Zero-shot visual question answering using knowledge graph. In *International Semantic Web Conference*, pages 146–162. Springer, 2021.

[48] Zhuo Chen, Yufeng Huang, Jiaoyan Chen, Yuxia Geng, Yin Fang, Jeff Z. Pan, Ningyu Zhang, and Wen Zhang. Lako: Knowledge-driven visual question answering via late knowledge-to-text injection. In *Proceedings of the 11th International Joint Conference on Knowledge Graphs*, IJCKG '22, page 20–29, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399876. doi: 10.1145/3579051.3579053. URL https://doi.org/10.1145/3579051.3579053.

[49] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829, 2023. doi: 10.1109/CVPR52729.2023.00276.

[50] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2829, 2023. doi: 10.1109/CVPR52729.2023.00276.

[51] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P.

Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

[52] Jaemin Cho, Seunghyun Yoon, Ajinkya Kale, Franck Dernoncourt, Trung Bui, and Mohit Bansal. Fine-grained image captioning with CLIP reward. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 517–527, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-naacl.39. URL https://aclanthology.org/2022.findings-naacl.39.

[53] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[54] Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4302–4310, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[55] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25 (70):1–53, 2024.

[56] Benjamin Clavié. Jacolbert and hard negatives, towards better japanese-first embeddings for retrieval: Early technical report. *arXiv preprint arXiv:2312.16144*, 2023.

[57] Alexandru Coca, Bo-Hsiang Tseng, Jinghong Chen, Weizhe Lin, Weixuan Zhang, Tisha Anders, and Bill Byrne. Grounding description-driven dialogue state trackers with knowledge-seeking turns. In Svetlana Stoyanchev, Shafiq Joty, David Schlangen, Ondrej Dusek, Casey Kennington, and Malihe Alikhani, editors, *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 444–456, Prague, Czechia, September 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.sigdial-1.42. URL https://aclanthology.org/2023.sigdial-1.42.

[58] Alexandru Coca, Bo-Hsiang Tseng, Weizhe Lin, and Bill Byrne. More robust schema-guided dialogue state tracking via tree-based paraphrase ranking. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1443–1454, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.106. URL https://aclanthology.org/2023.findings-eacl.106.

[59] Alexis Conneau and Guillaume Lample. *Cross-lingual language model pretraining*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[60] Yin Cui, Yang Song, Chen Sun, Andrew Howard, and Serge Belongie. Large scale fine-grained categorization and domain-specific transfer learning. (arXiv:1806.06193),

June 2018. doi: 10.48550/arXiv.1806.06193. URL http://arxiv.org/abs/1806.06193. arXiv:1806.06193 [cs].

[61] Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. Enabling multimodal generation on CLIP via vision-language knowledge distillation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2383–2395, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.187. URL https://aclanthology.org/2022.findings-acl.187.

[62] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36, 2024.

[63] Zhuyun Dai and Jamie Callan. Context-aware term weighting for first stage passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1533–1536, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401204. URL https://doi.org/10.1145/3397271.3401204.

[64] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[65] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/1feb87871436031bdc0f2beaa62a049b-Paper-Conference.pdf.

[66] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*. ACL, 2019. doi: 10.18653/v1/N19-1423. URL https://www.aclweb.org/anthology/N19-1423.

[67] Fernando Diaz. Regularizing ad hoc retrieval scores. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 672–679, 2005.

[68] Hanxing Ding, Liang Pang, Zihao Wei, Huawei Shen, and Xueqi Cheng. Retrieve only when it needs: Adaptive retrieval augmentation for hallucination mitigation in large language models. *arXiv preprint arXiv:2402.10612*, 2024.

[69] Haoyu Dong, Zhoujun Cheng, Xinyi He, Mengyu Zhou, Anda Zhou, Fan Zhou, Ao Liu, Shi Han, and Dongmei Zhang. Table pre-training: A survey on model architectures, pre-training objectives, and downstream tasks. In Lud De Raedt, editor, *Proceedings of the Thirty-First International Joint Conference on Artificial*

Intelligence, IJCAI-22, pages 5426–5435. International Joint Conferences on Artificial Intelligence Organization, 7 2022. doi: 10.24963/ijcai.2022/761. URL https://doi.org/10.24963/ijcai.2022/761. Survey Track.

[70] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. *Unified language model pre-training for natural language understanding and generation*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[71] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[72] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

[73] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar, Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: An embodied multimodal language model. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[74] Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

[75] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. GLM: General language model pretraining with autoregressive blank infilling. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.26. URL https://aclanthology.org/2022.acl-long.26.

[76] Miles Efron, Peter Organisciak, and Katrina Fenlon. Improving retrieval of short texts through document expansion. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 911–920, 2012.

[77] Julian Eisenschlos, Syrine Krichene, and Thomas Müller. Understanding tables with intermediate pre-training. In Trevor Cohn, Yulan He, and Yang Liu, editors, *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 281–296, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.27. URL https://aclanthology.org/2020.findings-emnlp.27.

[78] Julian Eisenschlos, Maharshi Gor, Thomas Müller, and William Cohen. MATE: Multi-view attention for table transformer efficiency. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7606–7619, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.600. URL https://aclanthology.org/2021.emnlp-main.600.

[79] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[80] Joel L Fagan. *Experiments in automatic phrase indexing for document retrieval: A comparison of syntactic and nonsyntactic methods*. Cornell University, 1988.

[81] Angela Fan, Claire Gardent, Chloé Braud, and Antoine Bordes. Augmenting transformers with knn-based composite memory for dialog. *Transactions of the Association for Computational Linguistics*, 9:82–99, 2021.

[82] Qingkai Fang and Yang Feng. Neural machine translation with phrase-level universal visual representations. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5687–5698, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.390. URL https://aclanthology.org/2022.acl-long.390.

[83] Manuel Faysse, Hugues Sibille, Tony Wu, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. *arXiv preprint arXiv:2407.01449*, 2024.

[84] William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39, 2022.

[85] Jiazhan Feng, Chongyang Tao, Xiubo Geng, Tao Shen, Can Xu, Guodong Long, Dongyan Zhao, and Daxin Jiang. Knowledge refinement via interaction between search engines and large language models. *arXiv preprint arXiv:2305.07402*, 2023.

[86] Yanlin Feng, Xinyue Chen, Bill Yuchen Lin, Peifeng Wang, Jun Yan, and Xiang Ren. Scalable multi-hop relational reasoning for knowledge-aware question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1295–1309, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.99. URL https://aclanthology.org/2020.emnlp-main.99.

[87] Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. *Advances in Neural Information Processing Systems*, 33:6616–6628, 2020.

[88] Feng Gao, Qing Ping, Govind Thattai, Aishwarya Reganti, Ying Nian Wu, and Prem Natarajan. 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, 2022.

[89] Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. Precise zero-shot dense retrieval without relevance labels. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.99. URL https://aclanthology.org/2023.acl-long.99.

[90] François Garderes, Maryam Ziaeefard, Baptiste Abeloos, and Freddy Lecue. Conceptbert: Concept-aware representation for visual question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 489–498, 2020.

[91] Arushi Goel, Basura Fernando, Frank Keller, and Hakan Bilen. Not all relations are equal: Mining informative labels for scene graph generation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15575–15585, 2022. doi: 10.1109/CVPR52688.2022.01515.

[92] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Handbook for Automatic Computation: Volume II: Linear Algebra*, pages 134–151. Springer, 1971.

[93] Google. Google lens: Image recognition and retrieval api. https://lens.google.com.

[94] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[95] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6904–6913, 2017.

[96] Liangke Gui, Borui Wang, Qiuyuan Huang, Alex Hauptmann, Yonatan Bisk, and Jianfeng Gao. Kat: A knowledge augmented transformer for vision-and-language. *arXiv preprint arXiv:2112.08614*, 2021.

[97] Dalu Guo, Chang Xu, and Dacheng Tao. Bilinear graph networks for visual question answering. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[98] Jiaxian Guo, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Boyang Li, Dacheng Tao, and Steven Hoi. From images to textual prompts: Zero-shot visual question answering with frozen large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10867–10877, June 2023.

[99] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org, 2020.

[100] Kiana Hajebi, Yasin Abbasi-Yadkori, Hossein Shahbazi, and Hong Zhang. Fast approximate nearest-neighbor search with k-nearest neighbor graph. In *Twenty-Second International Joint Conference on Artificial Intelligence*, 2011.

[101] Kailash A Hambarde and Hugo Proenca. Information retrieval: recent advances and beyond. *IEEE Access*, 2023.

[102] Zellig S Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

[103] Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. Efficient nearest neighbor language models. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.461. URL https://aclanthology.org/2021.emnlp-main.461.

[104] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[105] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. {DEBERTA}: {DECODING}-{enhanced} {bert} {with} {disentangled} {attention}. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=XPZIaotutsD.

[106] Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. TaPas: Weakly supervised table parsing via pre-training. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.398. URL https://aclanthology.org/2020.acl-main.398.

[107] Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Eisenschlos. Open domain question answering over tables via dense retrieval. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 512–519, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.43. URL https://aclanthology.org/2021.naacl-main.43.

[108] Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack W. Rae, and Laurent Sifre. Training compute-optimal large language models. In *Proceedings of the 36th International Conference on Neural Information*

Processing Systems, NIPS '22, Red Hook, NY, USA, 2024. Curran Associates Inc. ISBN 9781713871088.

[109] Sebastian Hofstätter, Omar Khattab, Sophia Althammer, Mete Sertkan, and Allan Hanbury. Introducing neural bag of whole-words with colberter: Contextualized late interactions using enhanced reduction. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 737–747, 2022.

[110] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

[111] Hanxu Hu, Hongyuan Lu, Huajian Zhang, Yun-Ze Song, Wai Lam, and Yue Zhang. Chain-of-symbol prompting elicits planning in large langauge models. *arXiv preprint arXiv:2305.10276*, 2023.

[112] Hexiang Hu, Yi Luan, Yang Chen, Urvashi Khandelwal, Mandar Joshi, Kenton Lee, Kristina Toutanova, and Ming-Wei Chang. Open-domain visual entity recognition: Towards recognizing millions of wikipedia entities. (arXiv:2302.11154), February 2023. URL http://arxiv.org/abs/2302.11154. arXiv:2302.11154 [cs].

[113] Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A Smith, and Jiebo Luo. Promptcap: Prompt-guided task-aware image captioning. *arXiv preprint arXiv:2211.09699*, 2022.

[114] Yushi Hu, Hang Hua, Zhengyuan Yang, Weijia Shi, Noah A. Smith, and Jiebo Luo. Promptcap: Prompt-guided image captioning for vqa with gpt-3. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2951–2963, 2023. doi: 10.1109/ICCV51070.2023.00277.

[115] Yushi Hu, Otilia Stretcu, Chun-Ta Lu, Krishnamurthy Viswanathan, Kenji Hata, Enming Luo, Ranjay Krishna, and Ariel Fuxman. Visual program distillation: Distilling tools and programmatic reasoning into vision-language models. *arXiv preprint arXiv:2312.03052*, 2023.

[116] Ziniu Hu, Ahmet Iscen, Chen Sun, Zirui Wang, Kai-Wei Chang, Yizhou Sun, Cordelia Schmid, David A. Ross, and Alireza Fathi. Reveal: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory. page 23369–23379, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Hu_REVEAL_Retrieval-Augmented_Visual-Language_Pre-Training_With_Multi-Source_Multimodal_Knowledge_Memory_CVPR_2023_paper.html.

[117] Ziniu Hu, Ahmet Iscen, Chen Sun, Kai-Wei Chang, Yizhou Sun, David Ross, Cordelia Schmid, and Alireza Fathi. Avis: Autonomous visual information seeking with large language model agent. *Advances in Neural Information Processing Systems*, 36, 2024.

[118] Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, et al. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *arXiv preprint arXiv:2311.05232*, 2023.

[119] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1233–1239, 2016.

[120] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[121] Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*, 2019. URL https://api.semanticscholar.org/CorpusID:210063976.

[122] Md Farhan Ishmam, Md Sakib Hossain Shovon, MF Mridha, and Nilanjan Dey. From image to language: A critical analysis of visual question answering (vqa) approaches, challenges, and opportunities. *Information Fusion*, page 102270, 2024.

[123] Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. Search-based neural structured learning for sequential question answering. In Regina Barzilay and Min-Yen Kan, editors, *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1167. URL https://aclanthology.org/P17-1167.

[124] Gautier Izacard and Edouard Grave. Leveraging passage retrieval with generative models for open domain question answering. In Paola Merlo, Jorg Tiedemann, and Reut Tsarfaty, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 874–880, Online, April 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.eacl-main.74. URL https://aclanthology.org/2021.eacl-main.74.

[125] Rolf Jagerman, Honglei Zhuang, Zhen Qin, Xuanhui Wang, and Michael Bendersky. Query expansion by prompting large language models. *arXiv preprint arXiv:2305.03653*, 2023.

[126] Aman Jain, Mayank Kothyari, Vishwajeet Kumar, Preethi Jyothi, Ganesh Ramakrishnan, and Soumen Chakrabarti. Select, substitute, search: A new benchmark for knowledge-augmented visual question answering. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2491–2498, 2021.

[127] Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C Park. Adaptive-rag: Learning to adapt retrieval-augmented large language models through question complexity. *arXiv preprint arXiv:2403.14403*, 2024.

[128] Kangyu Ji, Weizhe Lin, Yuqi Sun, Lin-Song Cui, Javad Shamsi, Yu-Hsien Chiang, Jiawei Chen, Elizabeth M. Tennyson, Linjie Dai, Qingbiao Li, Kyle Frohna, Miguel Anaya, Neil C. Greenham, and Samuel D. Stranks. Self-supervised deep learning for tracking degradation of perovskite light-emitting diodes with multispectral

imaging. *Nature Machine Intelligence*, 5(11):1225–1235, November 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00736-z. URL https://www.nature.com/articles/s42256-023-00736-z.

[129] Ziwei Ji, Himanshu Jain, Andreas Veit, Sashank J Reddi, Sadeep Jayasumana, Ankit Singh Rawat, Aditya Krishna Menon, Felix Yu, and Sanjiv Kumar. Efficient document ranking with learnable late interactions. *arXiv preprint arXiv:2406.17968*, 2024.

[130] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pages 4904–4916. PMLR, 2021.

[131] Yiren Jian, Chongyang Gao, and Soroush Vosoughi. Bootstrapping vision-language learning with decoupled language pre-training. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 57–72. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/002262941c9edfd472a79298b2ac5e17-Paper-Conference.pdf.

[132] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[133] Huaizu Jiang, Ishan Misra, Marcus Rohrbach, Erik Learned-Miller, and Xinlei Chen. 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, 2020.

[134] Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977, 2021. doi: 10.1162/tacl_a_00407. URL https://aclanthology.org/2021.tacl-1.57.

[135] Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.68. URL https://aclanthology.org/2022.naacl-main.68.

[136] Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. Active retrieval augmented generation. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.495. URL https://aclanthology.org/2023.emnlp-main.495.

[137] Matthew Jin, Syed Shahriar, Michele Tufano, Xin Shi, Shuai Lu, Neel Sundaresan, and Alexey Svyatkovskiy. Inferfix: End-to-end program repair with llms. In *Proceedings of the 31st ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, pages 1646–1656, 2023.

[138] Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[139] M. I. Jordan and T. M. Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. doi: 10.1126/science.aaa8415. URL https://www.science.org/doi/abs/10.1126/science.aaa8415.

[140] Nikhil Kandpal, Haikang Deng, Adam Roberts, Eric Wallace, and Colin Raffel. Large language models struggle to learn long-tail knowledge. In *International Conference on Machine Learning*, pages 15696–15707. PMLR, 2023.

[141] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[142] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[143] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 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, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.550. URL https://aclanthology.org/2020.emnlp-main.550.

[144] Asifullah Khan, Anabia Sohail, Umme Zahoora, and Aqsa Saeed Qureshi. A survey of the recent architectures of deep convolutional neural networks. *Artificial intelligence review*, 53:5455–5516, 2020.

[145] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=HklBjCEKvH.

[146] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450380164. doi: 10.1145/3397271.3401075. URL https://doi.org/10.1145/3397271.3401075.

[147] Gangwoo Kim, Sungdong Kim, Byeongguk Jeon, Joonsuk Park, and Jaewoo Kang. Tree of clarifications: Answering ambiguous questions with retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages

996–1009, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.63. URL https://aclanthology.org/2023.emnlp-main.63.

[148] Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pages 5583–5594. PMLR, 2021.

[149] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

[150] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, October 2023.

[151] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[152] Syrine Krichene, Thomas Müller, and Julian Eisenschlos. DoT: An efficient double transformer for NLP tasks with tables. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3273–3283, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.289. URL https://aclanthology.org/2021.findings-acl.289.

[153] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *Int. J. Comput. Vision*, 123(1):32–73, may 2017. ISSN 0920-5691. doi: 10.1007/s11263-016-0981-7. URL https://doi.org/10.1007/s11263-016-0981-7.

[154] Stefan Küchemann, Karina E Avila, Yavuz Dinc, Chiara Hortmann, Natalia Revenga, Verena Ruf, Niklas Stausberg, Steffen Steinert, Frank Fischer, Martin Fischer, et al. Are large multimodal foundation models all we need? on opportunities and challenges of these models in education. *EdArXiv*, 2024.

[155] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026.

[156] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466, 2019.

[157] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

[158] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[159] Daniel Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. *Advances in neural information processing systems*, 13, 2000.

[160] Jinhyuk Lee, Alexander Wettig, and Danqi Chen. Phrase retrieval learns passage retrieval, too. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3661–3672, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.297. URL https://aclanthology.org/2021.emnlp-main.297.

[161] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1612. URL https://aclanthology.org/P19-1612.

[162] Soohyeong Lee, Ju-Whan Kim, Youngmin Oh, and Joo Hyuk Jeon. Visual question answering over scene graph. In *2019 First International Conference on Graph Computing (GC)*, pages 45–50, 2019. doi: 10.1109/GC46384.2019.00015.

[163] Paul Lerner, Olivier Ferret, Camille Guinaudeau, Hervé Le Borgne, Romaric Besançon, José G Moreno, and Jesús Lovón Melgarejo. Viquae, a dataset for knowledge-based visual question answering about named entities. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3108–3120, 2022.

[164] Paul Lerner, Olivier Ferret, and Camille Guinaudeau. Multimodal inverse cloze task for knowledge-based visual question answering. In *European Conference on Information Retrieval*, pages 569–587. Springer, 2023.

[165] Paul Lerner, Olivier Ferret, and Camille Guinaudeau. Cross-modal retrieval for knowledge-based visual question answering. In *European Conference on Information Retrieval*, pages 421–438. Springer, 2024.

[166] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and

Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://aclanthology.org/2020.acl-main.703.

[167] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020.

[168] Guohao Li, Xin Wang, and Wenwu Zhu. Boosting visual question answering with context-aware knowledge aggregation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1227–1235, 2020.

[169] Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. *Advances in neural information processing systems*, 34:9694–9705, 2021.

[170] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022.

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

[172] Lei Li, Zhihui Xie, Mukai Li, Shunian Chen, Peiyi Wang, Liang Chen, Yazheng Yang, Benyou Wang, and Lingpeng Kong. Silkie: Preference distillation for large visual language models. *arXiv preprint arXiv:2312.10665*, 2023.

[173] Wei Li, Can Gao, Guocheng Niu, Xinyan Xiao, Hao Liu, Jiachen Liu, Hua Wu, and Haifeng Wang. UNIMO: Towards unified-modal understanding and generation via cross-modal contrastive learning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2592–2607, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.202. URL https://aclanthology.org/2021.acl-long.202.

[174] Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Mingjie Li, Wenjie Zhang, and Xuemin Lin. Approximate nearest neighbor search on high dimensional data—experiments, analyses, and improvement. *IEEE Transactions on Knowledge and Data Engineering*, 32(8):1475–1488, 2019.

[175] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision*, pages 121–137. Springer, 2020.

[176] Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. Monkey: Image resolution and text label are important things for large multi-modal models. *arXiv preprint arXiv:2311.06607*, 2023.

[177] Zujie Liang, Huang Hu, Can Xu, Chongyang Tao, Xiubo Geng, Yining Chen, Fan Liang, and Daxin Jiang. Maria: A visual experience powered conversational agent. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5596–5611, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.435. URL https://aclanthology.org/2021.acl-long.435.

[178] Bill Yuchen Lin, Xinyue Chen, Jamin Chen, and Xiang Ren. KagNet: Knowledge-aware graph networks for commonsense reasoning. 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 2829–2839, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1282. URL https://aclanthology.org/D19-1282.

[179] Kevin Lin, Ben Bogin, Mark Neumann, Jonathan Berant, and Matt Gardner. Grammar-based neural text-to-sql generation. *arXiv preprint arXiv:1905.13326*, 2019.

[180] Leroy Lin, Yujia Xie, Dongdong Chen, Yichong Xu, Chenguang Zhu, and Lu Yuan. REVIVE: Regional visual representation matters in knowledge-based visual question answering. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=wwyiEyK-G5D.

[181] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.

[182] Weizhe Lin. End-to-end multi-domain task-oriented dialogue systems. Master's thesis, 2021.

[183] Weizhe Lin and Bill Byrne. Retrieval augmented visual question answering with outside knowledge. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11238–11254, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.772. URL https://aclanthology.org/2022.emnlp-main.772.

[184] Weizhe Lin, Linjun Shou, Ming Gong, Jian Pei, Zhilin Wang, Bill Byrne, and Daxin Jiang. Combining unstructured content and knowledge graphs into recommendation datasets. In *KaRS@ RecSys*, pages 45–52, 2022.

[185] Weizhe Lin, Linjun Shou, Ming Gong, Jian Pei, Zhilin Wang, Bill Byrne, and Daxin Jiang. Transformer-empowered content-aware collaborative filtering. In *KaRS@ RecSys*, pages 53–64, 2022.

[186] Weizhe Lin, Rexhina Blloshmi, Bill Byrne, Adria de Gispert, and Gonzalo Iglesias. An inner table retriever for robust table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*

(Volume 1: Long Papers), pages 9909–9926, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.551. URL https://aclanthology.org/2023.acl-long.551.

[187] Weizhe Lin, Rexhina Blloshmi, Bill Byrne, Adria de Gispert, and Gonzalo Iglesias. LI-RAGE: Late interaction retrieval augmented generation with explicit signals for open-domain table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1557–1566, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.133. URL https://aclanthology.org/2023.acl-short.133.

[188] Weizhe Lin, Jinghong Chen, Jingbiao Mei, Alexandru Coca, and Bill Byrne. Fine-grained late-interaction multi-modal retrieval for retrieval augmented visual question answering. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=IWWWulAX7g.

[189] Weizhe Lin, Zhilin Wang, and Bill Byrne. FVQA 2.0: Introducing adversarial samples into fact-based visual question answering. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 149–157, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. URL https://aclanthology.org/2023.findings-eacl.11.

[190] Weizhe Lin, Jingbiao Mei, Jinghong Chen, and Bill Byrne. Preflmr: Scaling up fine-grained late-interaction multi-modal retrievers. In *Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[191] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

[192] Trond Linjordet. Neural (knowledge graph) question answering using synthetic training data. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 3245–3248, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3418505. URL https://doi.org/10.1145/3340531.3418505.

[193] Trond Linjordet and Krisztian Balog. Sanitizing synthetic training data generation for question answering over knowledge graphs. In *Proceedings of the 2020 ACM SIGIR on International Conference on Theory of Information Retrieval*, pages 121–128, 2020.

[194] Danyang Liu and Frank Keller. Detecting and grounding important characters in visual stories. In *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. ISBN 978-1-57735-880-0. doi: 10.1609/aaai.v37i11.26551. URL https://doi.org/10.1609/aaai.v37i11.26551.

[195] Danyang Liu, Mirella Lapata, and Frank Keller. Visual storytelling with question-answer plans. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5800–5813, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.386. URL https://aclanthology.org/2023.findings-emnlp.386.

[196] Hanchao Liu, Wenyuan Xue, Yifei Chen, Dapeng Chen, Xiutian Zhao, Ke Wang, Liping Hou, Rongjun Li, and Wei Peng. A survey on hallucination in large vision-language models. *arXiv preprint arXiv:2402.00253*, 2024.

[197] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023.

[198] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. (arXiv:2304.08485), April 2023. doi: 10.48550/arXiv.2304.08485. URL http://arxiv.org/abs/2304.08485. arXiv:2304.08485 [cs].

[199] Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9):1–35, 2023.

[200] Qian Liu, Bei Chen, Jiaqi Guo, Morteza Ziyadi, Zeqi Lin, Weizhu Chen, and Jian-Guang Lou. TAPEX: Table pre-training via learning a neural SQL executor. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=O50443AsCP.

[201] Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. 2024. URL arxiv.org/abs/2402.09353.

[202] Ye Liu, Semih Yavuz, Rui Meng, Dragomir Radev, Caiming Xiong, and Yingbo Zhou. Uni-parser: Unified semantic parser for question answering on knowledge base and database. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8858–8869, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.605. URL https://aclanthology.org/2022.emnlp-main.605.

[203] Yuqing Liu, Yu Wang, Lichao Sun, and Philip S Yu. Rec-gpt4v: Multimodal recommendation with large vision-language models. *arXiv preprint arXiv:2402.08670*, 2024.

[204] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[205] Junyu Lu, Ruyi Gan, Dixiang Zhang, Xiaojun Wu, Ziwei Wu, Renliang Sun, Jiaxing Zhang, Pingjian Zhang, and Yan Song. Lyrics: Boosting fine-grained language-vision alignment and comprehension via semantic-aware visual objects. *arXiv preprint arXiv:2312.05278*, 2023.

[206] Shuai Lu, Nan Duan, Hojae Han, Daya Guo, Seung-won Hwang, and Alexey Svyatkovskiy. ReACC: A retrieval-augmented code completion framework. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6227–6240, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.431. URL https://aclanthology.org/2022.acl-long.431.

[207] Yi Luan, Jacob Eisenstein, Kristina Toutanova, and Michael Collins. Sparse, dense, and attentional representations for text retrieval. *Transactions of the Association for Computational Linguistics*, 9:329–345, 2021. doi: 10.1162/tacl_a_00369. URL https://aclanthology.org/2021.tacl-1.20.

[208] Man Luo, Yankai Zeng, Pratyay Banerjee, and Chitta Baral. 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, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.517. URL https://aclanthology.org/2021.emnlp-main.517.

[209] Shangwen Lv, Daya Guo, Jingjing Xu, Duyu Tang, Nan Duan, Ming Gong, Linjun Shou, Daxin Jiang, Guihong Cao, and Songlin Hu. Graph-based reasoning over heterogeneous external knowledge for commonsense question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8449–8456, 2020.

[210] Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. Query rewriting in retrieval-augmented large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.322. URL https://aclanthology.org/2023.emnlp-main.322.

[211] Iain Mackie, Shubham Chatterjee, and Jeffrey Dalton. Generative and pseudo-relevant feedback for sparse, dense and learned sparse retrieval. *arXiv preprint arXiv:2305.07477*, 2023.

[212] Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. URL https://aclanthology.org/2023.acl-long.546.

[213] Kelong Mao, Zhicheng Dou, Fengran Mo, Jiewen Hou, Haonan Chen, and Hongjin Qian. Large language models know your contextual search intent: A prompting framework for conversational search. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1211–1225, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.86. URL https://aclanthology.org/2023.findings-emnlp.86.

[214] Kenneth Marino, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. 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, 2019.

[215] Kenneth Marino, Xinlei Chen, Devi Parikh, Abhinav Gupta, and Marcus Rohrbach. 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, 2021.

[216] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[217] Jingbiao Mei, Jinghong Chen, Weizhe Lin, Bill Byrne, and Marcus Tomalin. Improving hateful memes detection via learning hatefulness-aware embedding space through retrieval-guided contrastive learning. In *Proceedings of the 62st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[218] Thomas Mensink, Jasper Uijlings, Lluis Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. Encyclopedic vqa: Visual questions about detailed properties of fine-grained categories. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3113–3124, October 2023.

[219] Thomas Mensink, Jasper Uijlings, Lluis Castrejon, Arushi Goel, Felipe Cadar, Howard Zhou, Fei Sha, André Araujo, and Vittorio Ferrari. Encyclopedic vqa: Visual questions about detailed properties of fine-grained categories. (arXiv:2306.09224), July 2023. URL http://arxiv.org/abs/2306.09224. arXiv:2306.09224 [cs].

[220] Bertalan Meskó. The impact of multimodal large language models on health care's future. *J Med Internet Res*, 25:e52865, Nov 2023. ISSN 1438-8871. doi: 10.2196/52865. URL https://www.jmir.org/2023/1/e52865.

[221] Sewon Min, Danqi Chen, Hannaneh Hajishirzi, and Luke Zettlemoyer. A discrete hard EM approach for weakly supervised question answering. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan, editors, *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 2851–2864, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1284. URL https://aclanthology.org/D19-1284.

[222] Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wen-tau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. Nonparametric masked language modeling. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2097–2118, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.132. URL https://aclanthology.org/2023.findings-acl.132.

[223] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

[224] Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. Crosslingual generalization through multitask finetuning. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15991–16111, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.891. URL https://aclanthology.org/2023.acl-long.891.

[225] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. Generative representational instruction tuning. *arXiv preprint arXiv:2402.09906*, 2024.

[226] Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/c26820b8a4c1b3c2aa868d6d57e14a79-Paper.pdf.

[227] Medhini Narasimhan, Svetlana Lazebnik, and Alexander Schwing. Out of the box: Reasoning with graph convolution nets for factual visual question answering. *Advances in neural information processing systems*, 31, 2018.

[228] Franco Maria Nardini, Cosimo Rulli, and Rossano Venturini. Efficient multi-vector dense retrieval using bit vectors. *arXiv preprint arXiv:2404.02805*, 2024.

[229] Jianmo Ni, Gustavo Hernandez Abrego, Noah Constant, Ji Ma, Keith Hall, Daniel Cer, and Yinfei Yang. Sentence-t5: Scalable sentence encoders from pre-trained text-to-text models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Findings of the Association for Computational Linguistics: ACL 2022*, pages 1864–1874, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-acl.146. URL https://aclanthology.org/2022.findings-acl.146.

[230] Jianmo Ni, Chen Qu, Jing Lu, Zhuyun Dai, Gustavo Hernandez Abrego, Ji Ma, Vincent Zhao, Yi Luan, Keith Hall, Ming-Wei Chang, and Yinfei Yang. Large dual encoders are generalizable retrievers. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9844–9855, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.669. URL https://aclanthology.org/2022.emnlp-main.669.

[231] Sai Vidyaranya Nuthalapati, Ramraj Chandradevan, Eleonora Giunchiglia, Bowen Li, Maxime Kayser, Thomas Lukasiewicz, and Carl Yang. Lightweight visual question answering using scene graphs. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 3353–3357, New

York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3482218. URL https://doi.org/10.1145/3459637.3482218.

[232] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss,

Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2024.

[233] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

[234] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[235] Feifei Pan, Mustafa Canim, Michael Glass, Alfio Gliozzo, and Peter Fox. CLTR: An end-to-end, transformer-based system for cell-level table retrieval and table question answering. In Heng Ji, Jong C. Park, and Rui Xia, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 202–209, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-demo.24. URL https://aclanthology.org/2021.acl-demo.24.

[236] Feifei Pan, Mustafa Canim, Michael Glass, Alfio Gliozzo, and James Hendler. End-to-end table question answering via retrieval-augmented generation. *arXiv preprint arXiv:2203.16714*, 2022.

[237] Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In Chengqing Zong and Michael Strube, editors, *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1142. URL https://aclanthology.org/P15-1142.

[238] Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Derong Xu, Tong Xu, and Enhong Chen. Large language model based long-tail query rewriting in taobao search. In *Companion Proceedings of the ACM on Web Conference 2024*, WWW '24, page 20–28, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701726. doi: 10.1145/3589335.3648298. URL https://doi.org/10.1145/3589335.3648298.

[239] Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649, 2015.

[240] Yuankai Qi, Qi Wu, Peter Anderson, Xin Wang, William Yang Wang, Chunhua Shen, and Anton van den Hengel. Reverie: Remote embodied visual referring expression in

real indoor environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9982–9991, 2020.

[241] Yonggang Qiu and Hans-Peter Frei. Concept based query expansion. In *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 160–169, 1993.

[242] Chen Qu, Hamed Zamani, Liu Yang, W Bruce Croft, and Erik Learned-Miller. Passage retrieval for outside-knowledge visual question answering. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1753–1757, 2021.

[243] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. RocketQA: An optimized training approach to dense passage retrieval for open-domain question answering. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5835–5847, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.466. URL https://aclanthology.org/2021.naacl-main.466.

[244] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018. URL https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[245] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

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

[247] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

[248] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21 (140):1–67, 2020. URL http://jmlr.org/papers/v21/20-074.html.

[249] Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models. *Transactions of the Association for Computational Linguistics*, 11:1316–1331, 2023. doi: 10.1162/tacl_a_00605. URL https://aclanthology.org/2023.tacl-1.75.

[250] Rita Ramos, Desmond Elliott, and Bruno Martins. Retrieval-augmented image captioning. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3666–3681, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.266. URL https://aclanthology.org/2023.eacl-main.266.

[251] Jiahua Rao, Zifei Shan, Longpo Liu, Yao Zhou, and Yuedong Yang. Retrieval-based knowledge augmented vision language pre-training. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 5399–5409, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400701085. doi: 10.1145/3581783.3613848. URL https://doi.org/10.1145/3581783.3613848.

[252] Ruiyang Ren, Yuhao Wang, Yingqi Qu, Wayne Xin Zhao, Jing Liu, Hao Tian, Hua Wu, Ji-Rong Wen, and Haifeng Wang. Investigating the factual knowledge boundary of large language models with retrieval augmentation. *arXiv preprint arXiv:2307.11019*, 2023.

[253] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6):1137–1149, 2016.

[254] Adam Roberts, Colin Raffel, and Noam Shazeer. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.437. URL https://aclanthology.org/2020.emnlp-main.437.

[255] Stephen Robertson. Understanding inverse document frequency: on theoretical arguments for idf. *Journal of documentation*, 60(5):503–520, 2004.

[256] Stephen Robertson, Hugo Zaragoza, et al. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389, 2009.

[257] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[258] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

[259] Amir Saffari, Armin Oliya, Priyanka Sen, and Tom Ayoola. End-to-end entity resolution and question answering using differentiable knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages

4193–4200, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.345. URL https://aclanthology.org/2021.emnlp-main.345.

[260] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

[261] Gerard Salton and Christopher Buckley. Term-weighting approaches in automatic text retrieval. *Information processing & management*, 24(5):513–523, 1988.

[262] Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M Rush. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=9Vrb9D0WI4.

[263] Keshav Santhanam, Omar Khattab, Christopher Potts, and Matei Zaharia. Plaid: An efficient engine for late interaction retrieval. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 1747–1756, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450392365. doi: 10.1145/3511808.3557325. URL https://doi.org/10.1145/3511808.3557325.

[264] Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3715–3734, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.272. URL https://aclanthology.org/2022.naacl-main.272.

[265] Sara Sarto, Marcella Cornia, Lorenzo Baraldi, and Rita Cucchiara. Retrieval-augmented transformer for image captioning. In *Proceedings of the 19th international conference on content-based multimedia indexing*, pages 1–7, 2022.

[266] Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, Albert Webson, Pawan Sasanka Ammanamanchi, Thomas Wang, Benoît Sagot, Niklas Muennighoff, Albert Villanova del Moral, Olatunji Ruwase, Rachel Bawden, Stas Bekman, Angelina McMillan-Major, Iz Beltagy, Huu Nguyen, Lucile Saulnier, Samson Tan, Pedro Ortiz Suarez, Victor Sanh, Hugo Laurençon, Yacine Jernite, Julien Launay, Margaret Mitchell, Colin Raffel, Aaron Gokaslan, Adi Simhi, Aitor Soroa, Alham Fikri Aji, Amit Alfassy, Anna Rogers, Ariel Kreisberg Nitzav, Canwen Xu, Chenghao Mou, Chris Emezue, Christopher Klamm, Colin Leong, Daniel van Strien, David Ifeoluwa Adelani, Dragomir

Radev, Eduardo González Ponferrada, Efrat Levkovizh, Ethan Kim, Eyal Bar Natan, Francesco De Toni, Gérard Dupont, Germán Kruszewski, Giada Pistilli, Hady Elsahar, Hamza Benyamina, Hieu Tran, Ian Yu, Idris Abdulmumin, Isaac Johnson, Itziar Gonzalez-Dios, Javier de la Rosa, Jenny Chim, Jesse Dodge, Jian Zhu, Jonathan Chang, Jörg Frohberg, Joseph Tobing, Joydeep Bhattacharjee, Khalid Almubarak, Kimbo Chen, Kyle Lo, Leandro Von Werra, Leon Weber, Long Phan, Loubna Ben allal, Ludovic Tanguy, Manan Dey, Manuel Romero Muñoz, Maraim Masoud, María Grandury, Mario Šaško, Max Huang, Maximin Coavoux, Mayank Singh, Mike Tian-Jian Jiang, Minh Chien Vu, Mohammad A. Jauhar, Mustafa Ghaleb, Nishant Subramani, Nora Kassner, Nurulaqilla Khamis, Olivier Nguyen, Omar Espejel, Ona de Gibert, Paulo Villegas, Peter Henderson, Pierre Colombo, Priscilla Amuok, Quentin Lhoest, Rheza Harliman, Rishi Bommasani, Roberto Luis López, Rui Ribeiro, Salomey Osei, Sampo Pyysalo, Sebastian Nagel, Shamik Bose, Shamsuddeen Hassan Muhammad, Shanya Sharma, Shayne Longpre, Somaieh Nikpoor, Stanislav Silberberg, Suhas Pai, Sydney Zink, Tiago Timponi Torrent, Timo Schick, Tristan Thrush, Valentin Danchev, Vassilina Nikoulina, Veronika Laippala, Violette Lepercq, Vrinda Prabhu, Zaid Alyafeai, Zeerak Talat, Arun Raja, Benjamin Heinzerling, Chenglei Si, Davut Emre Taşar, Elizabeth Salesky, Sabrina J. Mielke, Wilson Y. Lee, Abheesht Sharma, Andrea Santilli, Antoine Chaffin, Arnaud Stiegler, Debajyoti Datta, Eliza Szczechla, Gunjan Chhablani, Han Wang, Harshit Pandey, Hendrik Strobelt, Jason Alan Fries, Jos Rozen, Leo Gao, Lintang Sutawika, M Saiful Bari, Maged S. Al-shaibani, Matteo Manica, Nihal Nayak, Ryan Teehan, Samuel Albanie, Sheng Shen, Srulik Ben-David, Stephen H. Bach, Taewoon Kim, Tali Bers, Thibault Fevry, Trishala Neeraj, Urmish Thakker, Vikas Raunak, Xiangru Tang, Zheng-Xin Yong, Zhiqing Sun, Shaked Brody, Yallow Uri, Hadar Tojarieh, Adam Roberts, Hyung Won Chung, Jaesung Tae, Jason Phang, Ofir Press, Conglong Li, Deepak Narayanan, Hatim Bourfoune, Jared Casper, Jeff Rasley, Max Ryabinin, Mayank Mishra, Minjia Zhang, Mohammad Shoeybi, Myriam Peyrounette, Nicolas Patry, Nouamane Tazi, Omar Sanseviero, Patrick von Platen, Pierre Cornette, Pierre François Lavallée, Rémi Lacroix, Samyam Rajbhandari, Sanchit Gandhi, Shaden Smith, Stéphane Requena, Suraj Patil, Tim Dettmers, Ahmed Baruwa, Amanpreet Singh, Anastasia Cheveleva, Anne-Laure Ligozat, Arjun Subramonian, Aurélie Névéol, Charles Lovering, Dan Garrette, Deepak Tunuguntla, Ehud Reiter, Ekaterina Taktasheva, Ekaterina Voloshina, Eli Bogdanov, Genta Indra Winata, Hailey Schoelkopf, Jan-Christoph Kalo, Jekaterina Novikova, Jessica Zosa Forde, Jordan Clive, Jungo Kasai, Ken Kawamura, Liam Hazan, Marine Carpuat, Miruna Clinciu, Najoung Kim, Newton Cheng, Oleg Serikov, Omer Antverg, Oskar van der Wal, Rui Zhang, Ruochen Zhang, Sebastian Gehrmann, Shachar Mirkin, Shani Pais, Tatiana Shavrina, Thomas Scialom, Tian Yun, Tomasz Limisiewicz, Verena Rieser, Vitaly Protasov, Vladislav Mikhailov, Yada Pruksachatkun, Yonatan Belinkov, Zachary Bamberger, Zdeněk Kasner, Alice Rueda, Amanda Pestana, Amir Feizpour, Ammar Khan, Amy Faranak, Ana Santos, Anthony Hevia, Antigona Unldreaj, Arash Aghagol, Arezoo Abdollahi, Aycha Tammour, Azadeh HajiHosseini, Bahareh Behroozi, Benjamin Ajibade, Bharat Saxena, Carlos Muñoz Ferrandis, Daniel McDuff, Danish Contractor, David Lansky, Davis David, Douwe Kiela, Duong A. Nguyen, Edward Tan, Emi Baylor, Ezinwanne Ozoani, Fatima Mirza, Frankline Ononiwu, Habib Rezanejad, Hessie Jones, Indrani Bhattacharya, Irene Solaiman, Irina Sedenko, Isar Nejadgholi, Jesse Passmore, Josh Seltzer, Julio Bonis Sanz, Livia Dutra, Mairon Samagaio, Maraim Elbadri, Margot Mieskes, Marissa Gerchick, Martha Akinlolu, Michael McKenna, Mike

Qiu, Muhammed Ghauri, Mykola Burynok, Nafis Abrar, Nazneen Rajani, Nour Elkott, Nour Fahmy, Olanrewaju Samuel, Ran An, Rasmus Kromann, Ryan Hao, Samira Alizadeh, Sarmad Shubber, Silas Wang, Sourav Roy, Sylvain Viguier, Thanh Le, Tobi Oyebade, Trieu Le, Yoyo Yang, Zach Nguyen, Abhinav Ramesh Kashyap, Alfredo Palasciano, Alison Callahan, Anima Shukla, Antonio Miranda-Escalada, Ayush Singh, Benjamin Beilharz, Bo Wang, Caio Brito, Chenxi Zhou, Chirag Jain, Chuxin Xu, Clémentine Fourrier, Daniel León Periñán, Daniel Molano, Dian Yu, Enrique Manjavacas, Fabio Barth, Florian Fuhrimann, Gabriel Altay, Giyaseddin Bayrak, Gully Burns, Helena U. Vrabec, Imane Bello, Ishani Dash, Jihyun Kang, John Giorgi, Jonas Golde, Jose David Posada, Karthik Rangasai Sivaraman, Lokesh Bulchandani, Lu Liu, Luisa Shinzato, Madeleine Hahn de Bykhovetz, Maiko Takeuchi, Marc Pàmies, Maria A Castillo, Marianna Nezhurina, Mario Sänger, Matthias Samwald, Michael Cullan, Michael Weinberg, Michiel De Wolf, Mina Mihaljcic, Minna Liu, Moritz Freidank, Myungsun Kang, Natasha Seelam, Nathan Dahlberg, Nicholas Michio Broad, Nikolaus Muellner, Pascale Fung, Patrick Haller, Ramya Chandrasekhar, Renata Eisenberg, Robert Martin, Rodrigo Canalli, Rosaline Su, Ruisi Su, Samuel Cahyawijaya, Samuele Garda, Shlok S Deshmukh, Shubhanshu Mishra, Sid Kiblawi, Simon Ott, Sinee Sangaroonsiri, Srishti Kumar, Stefan Schweter, Sushil Bharati, Tanmay Laud, Théo Gigant, Tomoya Kainuma, Wojciech Kusa, Yanis Labrak, Yash Shailesh Bajaj, Yash Venkatraman, Yifan Xu, Yingxin Xu, Yu Xu, Zhe Tan, Zhongli Xie, Zifan Ye, Mathilde Bras, Younes Belkada, and Thomas Wolf. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022.

[267] Noel Schäfer, Sebastian Künzel, Tanja Munz-Körner, Pascal Tilli, Sandeep Vidyapu, Ngoc Thang Vu, and Daniel Weiskopf. Visual analysis of scene-graph-based visual question answering. In *Proceedings of the 16th International Symposium on Visual Information Communication and Interaction*, VINCI '23, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9798400707513. doi: 10.1145/3615522.3615547. URL https://doi.org/10.1145/3615522.3615547.

[268] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[269] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, Patrick Schramowski, Srivatsa Kundurthy, Katherine Crowson, Ludwig Schmidt, Robert Kaczmarczyk, and Jenia Jitsev. Laion-5b: An open large-scale dataset for training next generation image-text models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 25278–25294. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/a1859debfb3b59d094f3504d5ebb6c25-Paper-Datasets_and_Benchmarks.pdf.

[270] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[271] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*, volume 39. Cambridge University Press Cambridge, 2008.

[272] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark fornbsp;visual question answering using world knowledge. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VIII*, page 146–162, Berlin, Heidelberg, 2022. Springer-Verlag. ISBN 978-3-031-20073-1. doi: 10.1007/978-3-031-20074-8_9. URL https://doi.org/10.1007/978-3-031-20074-8_9.

[273] Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-okvqa: A benchmark for visual question answering using world knowledge. *arXiv preprint arXiv:2206.01718*, 2022.

[274] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[275] Sanket Shah, Anand Mishra, Naganand Yadati, and Partha Pratim Talukdar. Kvqa: Knowledge-aware visual question answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8876–8884, July 2019. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v33i01.33018876.

[276] Hao Shao, Shengju Qian, Han Xiao, Guanglu Song, Zhuofan Zong, Letian Wang, Yu Liu, and Hongsheng Li. Visual cot: Unleashing chain-of-thought reasoning in multi-modal language models. *arXiv preprint arXiv:2403.16999*, 2024.

[277] Zhenwei Shao, Zhou Yu, Meng Wang, and Jun Yu. 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, 2023.

[278] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 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)*, page 2556–2565, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1238. URL https://aclanthology.org/P18-1238.

[279] Lei Shen, Haolan Zhan, Xin Shen, Yonghao Song, and Xiaofang Zhao. Text is not enough: Integrating visual impressions into open-domain dialogue generation. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4287–4296, 2021.

[280] Tao Shen, Guodong Long, Xiubo Geng, Chongyang Tao, Tianyi Zhou, and Daxin Jiang. Large language models are strong zero-shot retriever. *arXiv preprint arXiv:2304.14233*, 2023.

[281] Yunhang Shen, Chaoyou Fu, Peixian Chen, Mengdan Zhang, Ke Li, Xing Sun, Yunsheng Wu, Shaohui Lin, and Rongrong Ji. Aligning and prompting everything all at once for universal visual perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13193–13203, 2024.

[282] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[283] Ensheng Shi, Yanlin Wang, Wei Tao, Lun Du, Hongyu Zhang, Shi Han, Dongmei Zhang, and Hongbin Sun. RACE: Retrieval-augmented commit message generation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5520–5530, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.372. URL https://aclanthology.org/2022.emnlp-main.372.

[284] Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*, 2023.

[285] Zhan Shi, Hui Liu, Martin Renqiang Min, Christopher Malon, Li Erran Li, and Xiaodan Zhu. Retrieval, analogy, and composition: A framework for compositional generalization in image captioning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1990–2000, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.171. URL https://aclanthology.org/2021.findings-emnlp.171.

[286] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019.

[287] Amanpreet Singh, Ronghang Hu, Vedanuj Goswami, Guillaume Couairon, Wojciech Galuba, Marcus Rohrbach, and Douwe Kiela. Flava: A foundational language and vision alignment model. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15617–15629, 2022. doi: 10.1109/CVPR52688.2022.01519.

[288] Robyn Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

[289] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2443–2449, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3463257. URL https://doi.org/10.1145/3404835.3463257.

[290] Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 2443–2449, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3463257. URL https://doi.org/10.1145/3404835.3463257.

[291] Krishna Srinivasan, Karthik Raman, Anupam Samanta, Lingrui Liao, Luca Bertelli, and Michael Bendersky. QUILL: Query intent with large language models using retrieval augmentation and multi-stage distillation. In Yunyao Li and Angeliki Lazaridou, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 492–501, Abu Dhabi, UAE, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-industry.50. URL https://aclanthology.org/2022.emnlp-industry.50.

[292] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023.

[293] Zhiqing Sun, Sheng Shen, Shengcao Cao, Haotian Liu, Chunyuan Li, Yikang Shen, Chuang Gan, Liang-Yan Gui, Yu-Xiong Wang, Yiming Yang, et al. Aligning large multimodal models with factually augmented rlhf. *arXiv preprint arXiv:2309.14525*, 2023.

[294] Hao Tan and Mohit Bansal. 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, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1514. URL https://aclanthology.org/D19-1514.

[295] Niket Tandon, Gerard De Melo, and Gerhard Weikum. Webchild 2.0: Fine-grained commonsense knowledge distillation. In *Proceedings of ACL 2017, System Demonstrations*, pages 115–120, 2017.

[296] K. Tang, Y. Niu, J. Huang, J. Shi, and H. Zhang. Unbiased scene graph generation from biased training. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3713–3722, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. doi: 10.1109/CVPR42600.2020.00377. URL https://doi.ieeecomputersociety.org/10.1109/CVPR42600.2020.00377.

[297] Kaihua Tang, Hanwang Zhang, Baoyuan Wu, Wenhan Luo, and Wei Liu. Learning to compose dynamic tree structures for visual contexts. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6612–6621, 2019. doi: 10.1109/CVPR.2019.00678.

[298] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

[299] Yi Tay, Mostafa Dehghani, Vinh Q. Tran, Xavier Garcia, Jason Wei, Xuezhi Wang, Hyung Won Chung, Dara Bahri, Tal Schuster, Steven Zheng, Denny Zhou, Neil Houlsby, and Donald Metzler. UL2: Unifying language learning paradigms. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6ruVLB727MC.

[300] Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini:

a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[301] Anthony Meng Huat Tiong, Junnan Li, Boyang Li, Silvio Savarese, and Steven C.H. Hoi. Plug-and-play VQA: Zero-shot VQA by conjoining large pretrained models with zero training. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 951–967, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.67. URL https://aclanthology.org/2022.findings-emnlp.67.

[302] SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*, 2024.

[303] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[304] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[305] Priyansh Trivedi, Gaurav Maheshwari, Mohnish Dubey, and Jens Lehmann. Lc-quad: A corpus for complex question answering over knowledge graphs. In *International Semantic Web Conference*, pages 210–218. Springer, 2017.

[306] Maria Tsimpoukelli, Jacob Menick, Serkan Cabi, SM Eslami, Oriol Vinyals, and Felix Hill. Multimodal few-shot learning with frozen language models. *arXiv preprint arXiv:2106.13884*, 2021.

[307] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[308] Ellen M Voorhees. Query expansion using lexical-semantic relations. In *SIGIR'94: Proceedings of the Seventeenth Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, organised by Dublin City University*, pages 61–69. Springer, 1994.

[309] Fei Wang, Kexuan Sun, Muhao Chen, Jay Pujara, and Pedro Szekely. Retrieving complex tables with multi-granular graph representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 1472–1482, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450380379. doi: 10.1145/3404835.3462909. URL https://doi.org/10.1145/3404835.3462909.

[310] Liang Wang, Nan Yang, and Furu Wei. Query2doc: Query expansion with large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.585. URL https://aclanthology.org/2023.emnlp-main.585.

[311] Peng Wang, Qi Wu, Chunhua Shen, Anthony Dick, and Anton Van Den Hengel. Fvqa: Fact-based visual question answering. *IEEE transactions on pattern analysis and machine intelligence*, 40(10):2413–2427, 2017.

[312] Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, et al. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19175–19186, 2023.

[313] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

[314] Yile Wang, Peng Li, Maosong Sun, and Yang Liu. Self-knowledge guided retrieval augmentation for large language models. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10303–10315, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.691. URL https://aclanthology.org/2023.findings-emnlp.691.

[315] Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1780–1790, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383325. doi: 10.1145/3447548.3467434. URL https://doi.org/10.1145/3447548.3467434.

[316] Zhiruo Wang, Zhengbao Jiang, Eric Nyberg, and Graham Neubig. Table retrieval may not necessitate table-specific model design. In Wenhu Chen, Xinyun Chen, Zhiyu Chen, Ziyu Yao, Michihiro Yasunaga, Tao Yu, and Rui Zhang, editors, *Proceedings of the Workshop on Structured and Unstructured Knowledge Integration (SUKI)*, pages 36–46, Seattle, USA, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.suki-1.5. URL https://aclanthology.org/2022.suki-1.5.

[317] Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. Chain-of-table: Evolving tables in the reasoning chain for table understanding. *arXiv preprint arXiv:2401.04398*, 2024.

[318] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In

International Conference on Learning Representations, 2022. URL https://openreview.net/forum?id=GUrhfTuf_3.

[319] Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. Uniir: Training and benchmarking universal multimodal information retrievers. *arXiv preprint arXiv:2311.17136*, 2023.

[320] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf.

[321] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval. (arXiv:2004.01804), November 2020. doi: 10.48550/arXiv.2004.01804. URL http://arxiv.org/abs/2004.01804. arXiv:2004.01804 [cs].

[322] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 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, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6.

[323] SK Michael Wong, Wojciech Ziarko, and Patrick CN Wong. Generalized vector spaces model in information retrieval. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 18–25, 1985.

[324] Jialin Wu and Ray Mooney. Breaking down questions for outside-knowledge VQA. In *Submitted to The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=ILYX-vQnwe_. Conference Withdrawn Submission.

[325] Jialin Wu and Raymond Mooney. Entity-focused dense passage retrieval for outside-knowledge visual question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8061–8072, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.551. URL https://aclanthology.org/2022.emnlp-main.551.

[326] Jialin Wu, Jiasen Lu, Ashish Sabharwal, and Roozbeh Mottaghi. Multi-modal answer validation for knowledge-based vqa. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2712–2721, 2022.

[327] X.ai. Open release of grok-1, March 17 2024. URL https://x.ai/blog/grok-os. Accessed on 2024-05-17.

[328] Alexandros Xenos, Themos Stafylakis, Ioannis Patras, and Georgios Tzimiropoulos. A simple baseline for knowledge-based visual question answering. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14871–14877, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.919. URL https://aclanthology.org/2023.emnlp-main.919.

[329] Hanguang Xiao, Feizhong Zhou, Xingyue Liu, Tianqi Liu, Zhipeng Li, Xin Liu, and Xiaoxuan Huang. A comprehensive survey of large language models and multimodal large language models in medicine. *arXiv preprint arXiv:2405.08603*, 2024.

[330] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=zeFrfgyZln.

[331] Chunpu Xu, Min Yang, Chengming Li, Ying Shen, Xiang Ao, and Ruifeng Xu. Imagine, reason and write: Visual storytelling with graph knowledge and relational reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3022–3029, 2021.

[332] Danfei Xu, Yuke Zhu, Christopher B. Choy, and Li Fei-Fei. Scene graph generation by iterative message passing. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3097–3106, 2017. doi: 10.1109/CVPR.2017.330.

[333] Xiaojun Xu, Chang Liu, and Dawn Song. SQLNet: Generating structured queries from natural language without reinforcement learning, 2018. URL https://openreview.net/forum?id=SkYibHlRb.

[334] Xin Xu, Yue Liu, Panupong Pasupat, Mehran Kazemi, et al. In-context learning with retrieved demonstrations for language models: A survey. *arXiv preprint arXiv:2401.11624*, 2024.

[335] Yilun Xu, Shengjia Zhao, Jiaming Song, Russell Stewart, and Stefano Ermon. A theory of usable information under computational constraints. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=r1eBeyHFDH.

[336] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.41. URL https://aclanthology.org/2021.naacl-main.41.

[337] Ikuya Yamada, Akari Asai, and Hannaneh Hajishirzi. Efficient passage retrieval with hashing for open-domain question answering. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 979–986, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.123. URL https://aclanthology.org/2021.acl-short.123.

[338] Guangyu Yang, Jinghong Chen, Weizhe Lin, and Bill Byrne. Direct preference optimization for neural machine translation with minimum Bayes risk decoding. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 391–398, Mexico City, Mexico, June 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.naacl-short.34.

[339] Jingfeng Yang, Aditya Gupta, Shyam Upadhyay, Luheng He, Rahul Goel, and Shachi Paul. TableFormer: Robust transformer modeling for table-text encoding. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 528–537, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.40. URL https://aclanthology.org/2022.acl-long.40.

[340] Yue Yang, Wenlin Yao, Hongming Zhang, Xiaoyang Wang, Dong Yu, and Jianshu Chen. Z-LaVI: Zero-shot language solver fueled by visual imagination. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1186–1203, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.78. URL https://aclanthology.org/2022.emnlp-main.78.

[341] Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Yumao Lu, Zicheng Liu, and Lijuan Wang. 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, 2022.

[342] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. *XLNet: generalized autoregressive pretraining for language understanding*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[343] Zhuolin Yang, Wei Ping, Zihan Liu, Vijay Korthikanti, Weili Nie, De-An Huang, Linxi Fan, Zhiding Yu, Shiyi Lan, Bo Li, Mohammad Shoeybi, Ming-Yu Liu, Yuke Zhu, Bryan Catanzaro, Chaowei Xiao, and Anima Anandkumar. Re-ViLM: Retrieval-augmented visual language model for zero and few-shot image captioning. In Houda Bouamor, Juan Pino, and Kalika Bali, editors, *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 11844–11857, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.793. URL https://aclanthology.org/2023.findings-emnlp.793.

[344] Zhuoqian Yang, Zengchang Qin, Jing Yu, and Tao Wan. Prior visual relationship reasoning for visual question answering. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 1411–1415, 2020. doi: 10.1109/ICIP40778.2020.9190771.

[345] Zonglin Yang, Xinya Du, Erik Cambria, and Claire Cardie. End-to-end case-based reasoning for commonsense knowledge base completion. In Andreas Vlachos and Isabelle Augenstein, editors, *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 3509–3522, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.255. URL https://aclanthology.org/2023.eacl-main.255.

[346] Lewei Yao, Runhui Huang, Lu Hou, Guansong Lu, Minzhe Niu, Hang Xu, Xiaodan Liang, Zhenguo Li, Xin Jiang, and Chunjing Xu. FILIP: Fine-grained interactive language-image pre-training. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=cpDhcsEDC2.

[347] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/271db9922b8d1f4dd7aaef84ed5ac703-Paper-Conference.pdf.

[348] Yao Yao, Zuchao Li, and Hai Zhao. Beyond chain-of-thought, effective graph-of-thought reasoning in large language models. *arXiv preprint arXiv:2305.16582*, 2023.

[349] Michihiro Yasunaga, Armen Aghajanyan, Weijia Shi, Rich James, Jure Leskovec, Percy Liang, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. Retrieval-augmented multimodal language modeling. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[350] Xi Ye, Semih Yavuz, Kazuma Hashimoto, Yingbo Zhou, and Caiming Xiong. RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6032–6043, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.417. URL https://aclanthology.org/2022.acl-long.417.

[351] Yuyang Ye, Zhi Zheng, Yishan Shen, Tianshu Wang, Hengruo Zhang, Peijun Zhu, Runlong Yu, Kai Zhang, and Hui Xiong. Harnessing multimodal large language models for multimodal sequential recommendation. *arXiv preprint arXiv:2408.09698*, 2024.

[352] Da Yin, Feng Gao, Govind Thattai, Michael Johnston, and Kai-Wei Chang. Givl: Improving geographical inclusivity of vision-language models with pre-training methods. page 10951–10961, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Yin_GIVL_Improving_Geographical_Inclusivity_of_Vision-Language_Models_With_Pre-Training_Methods_CVPR_2023_paper.html.

[353] Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. TaBERT: Pretraining for joint understanding of textual and tabular data. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.745. URL https://aclanthology.org/2020.acl-main.745.

[354] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

[355] Chi Yu, Guang Yang, Xiang Chen, Ke Liu, and Yanlin Zhou. Bashexplainer: Retrieval-augmented bash code comment generation based on fine-tuned codebert. In *2022 IEEE International Conference on Software Maintenance and Evolution (ICSME)*, pages 82–93. IEEE, 2022.

[356] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://api.semanticscholar.org/CorpusID:248512473.

[357] Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1425. URL https://aclanthology.org/D18-1425.

[358] Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, bailin wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, richard socher, and Caiming Xiong. Gra{pp}a: Grammar-augmented pre-training for table semantic parsing. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=kyaIeYj4zZ.

[359] Tao Yu, Chien-Sheng Wu, Xi Victoria Lin, bailin wang, Yi Chern Tan, Xinyi Yang, Dragomir Radev, richard socher, and Caiming Xiong. Gra{pp}a: Grammar-augmented pre-training for table semantic parsing. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=kyaIeYj4zZ.

[360] Tianyu Yu, Yuan Yao, Haoye Zhang, Taiwen He, Yifeng Han, Ganqu Cui, Jinyi Hu, Zhiyuan Liu, Hai-Tao Zheng, Maosong Sun, et al. Rlhf-v: Towards trustworthy mllms via behavior alignment from fine-grained correctional human feedback. *arXiv preprint arXiv:2312.00849*, 2023.

[361] Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. Generate rather than retrieve: Large language models are strong context generators. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=fB0hRu9GZUS.

[362] Zhou Yu, Jun Yu, Chenchao Xiang, Jianping Fan, and Dacheng Tao. Beyond bilinear: Generalized multimodal factorized high-order pooling for visual question answering. *IEEE transactions on neural networks and learning systems*, 29(12):5947–5959, 2018.

[363] Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 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, 2019.

[364] Yuqian Yuan, Wentong Li, Jian Liu, Dongqi Tang, Xinjie Luo, Chi Qin, Lei Zhang, and Jianke Zhu. Osprey: Pixel understanding with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 28202–28211, June 2024.

[365] Zheng Yuan, Qiao Jin, Chuanqi Tan, Zhengyun Zhao, Hongyi Yuan, Fei Huang, and Songfang Huang. Ramm: Retrieval-augmented biomedical visual question answering with multi-modal pre-training. *Proceedings of the 31st ACM International Conference on Multimedia*, 2023.

[366] Hamed Zamani, Mostafa Dehghani, W Bruce Croft, Erik Learned-Miller, and Jaap Kamps. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM international conference on information and knowledge management*, pages 497–506, 2018.

[367] Chengxiang Zhai and John Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, 2001.

[368] Duzhen Zhang, Yahan Yu, Chenxing Li, Jiahua Dong, Dan Su, Chenhui Chu, and Dong Yu. Mm-llms: Recent advances in multimodal large language models. *arXiv preprint arXiv:2401.13601*, 2024.

[369] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[370] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. Vinvl: Revisiting visual representations in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5579–5588, 2021.

[371] Yutong Zhang, Yi Pan, Tianyang Zhong, Peixin Dong, Kangni Xie, Yuxiao Liu, Hanqi Jiang, Zhengliang Liu, Shijie Zhao, Tuo Zhang, et al. Potential of multimodal large language models for data mining of medical images and free-text reports. *arXiv preprint arXiv:2407.05758*, 2024.

[372] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=5NTt8GFjUHkr.

[373] Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023.

[374] Nan Zhao, Haoran Li, Youzheng Wu, Xiaodong He, and Bowen Zhou. The JDDC 2.0 corpus: A large-scale multimodal multi-turn chinese dialogue dataset for e-commerce customer service. *CoRR*, abs/2109.12913, 2021. URL https://arxiv.org/abs/2109.12913.

[375] Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, and Bin Cui. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*, 2024.

[376] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. Dense text retrieval based on pretrained language models: A survey. *ACM Transactions on Information Systems*, 42(4):1–60, 2024.

[377] Xufeng Zhao, Mengdi Li, Wenhao Lu, Cornelius Weber, Jae Hee Lee, Kun Chu, and Stefan Wermter. Enhancing Zero-Shot Chain-of-Thought Reasoning in Large Language Models through Logic. In *2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024), Turin, Italy*, May 2024.

[378] Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibei Yang. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 5168–5191. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/108030643e640ac050e0ed5e6aace48f-Paper-Conference.pdf.

[379] Guoqing Zheng and Jamie Callan. Learning to reweight terms with distributed representations. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*, pages 575–584, 2015.

[380] Kaizhi Zheng, Xuehai He, and Xin Eric Wang. Minigpt-5: Interleaved vision-and-language generation via generative vokens. *arXiv preprint arXiv:2310.02239*, 2023.

[381] Victor Zhong, Caiming Xiong, and Richard Socher. Seq2sql: Generating structured queries from natural language using reinforcement learning. *CoRR*, abs/1709.00103, 2017. URL http://arxiv.org/abs/1709.00103.

[382] Zexuan Zhong, Tao Lei, and Danqi Chen. Training language models with memory augmentation. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5657–5673, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.382. URL https://aclanthology.org/2022.emnlp-main.382.

[383] Fan Zhou, Mengkang Hu, Haoyu Dong, Zhoujun Cheng, Shi Han, and Dongmei Zhang. TaCube: Pre-computing Data Cubes for Answering Numerical-Reasoning Questions over Tabular Data. In *Proceedings of the 2022 Conference on Empirical Methods in*

Natural Language Processing (EMNLP), Abu Dhabi, United Arab Emirates, dec 2022. Association for Computational Linguistics. URL https://preview.aclanthology.org/emnlp-22-ingestion/2022.emnlp-main.145/.

[384] Junjie Zhou, Zheng Liu, Shitao Xiao, Bo Zhao, and Yongping Xiong. Vista: Visualized text embedding for universal multi-modal retrieval. *arXiv preprint arXiv:2406.04292*, 2024.

[385] Mingyang Zhou, Grace Luo, Anna Rohrbach, and Zhou Yu. Focus! relevant and sufficient context selection for news image captioning. In Yoav Goldberg, Zornitsa Kozareva, and Yue Zhang, editors, *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6078–6088, Abu Dhabi, United Arab Emirates, December 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.findings-emnlp.450. URL https://aclanthology.org/2022.findings-emnlp.450.

[386] Yucheng Zhou and Guodong Long. Style-aware contrastive learning for multi-style image captioning. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 2257–2267, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.169. URL https://aclanthology.org/2023.findings-eacl.169.

[387] Yucheng Zhou, Xiubo Geng, Tao Shen, Chongyang Tao, Guodong Long, Jian-Guang Lou, and Jianbing Shen. Thread of thought unraveling chaotic contexts. *arXiv preprint arXiv:2311.08734*, 2023.

[388] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

[389] Wanrong Zhu, An Yan, Yujie Lu, Wenda Xu, Xin Wang, Miguel Eckstein, and William Yang Wang. Visualize before you write: Imagination-guided open-ended text generation. In Andreas Vlachos and Isabelle Augenstein, editors, *Findings of the Association for Computational Linguistics: EACL 2023*, pages 78–92, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-eacl.5. URL https://aclanthology.org/2023.findings-eacl.5.

[390] Zihao Zhu, Jing Yu, Yujing Wang, Yajing Sun, Yue Hu, and Qi Wu. Mucko: Multi-layer cross-modal knowledge reasoning for fact-based visual question answering. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 1097–1103. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/153. URL https://doi.org/10.24963/ijcai.2020/153. Main track.

[391] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In Sheng Li, Maosong Sun, Yang Liu, Hua Wu, Kang Liu, Wanxiang Che, Shizhu He, and Gaoqi Rao, editors, *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China, August 2021. Chinese Information Processing Society of China. URL https://aclanthology.org/2021.ccl-1.108.

[392] Yongshuo Zong, Ondrej Bohdal, and Timothy Hospedales. Vl-icl bench: The devil in the details of benchmarking multimodal in-context learning. *arXiv preprint arXiv:2403.13164*, 2024.

[393] Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Hospedales Timothy. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. *arXiv preprint arXiv:2402.02207*, 2024.

[394] Guido Zuccon, Bevan Koopman, Peter Bruza, and Leif Azzopardi. Integrating and evaluating neural word embeddings in information retrieval. In *Proceedings of the 20th Australasian document computing symposium*, pages 1–8, 2015.

Appendix A

Appendix for Chapter 5

A.1 Data Statistics of OK-VQA

Table A.1 shows the data statistics of the OK-VQA dataset. Table A.2 displays the number of passages in the document collections used for evaluating the retrieval systems. Note that the WIT corpus is introduced in Appendix A.4, which is used for investigating the retrieval of multi-modal documents.

Table A.1 OK-VQA dataset statistics.

Category	Number
Train questions	9,009
Valid questions	5,046
Images	14,055

Table A.2 Data statistics of document collections used in retrieval.

Corpus	# of passages
GS for OK-VQA [208]	168,306
Wikipedia for OK-VQA	114,637
WIT for OK-VQA (Appendix A.4)	87,419

A.2 Training and Hyperparameter Details

We use pre-trained checkpoints from `huggingface`.

We use `ColBERTv2` and `openai/clip-vit-base-patch32` to initialise the text-based retriever and vision encoder. For the DPR baseline, we use `facebook/dpr-single-nq-base` to initialise the retriever. In answer generation, we use `t5-large` and `Salesforce/blip2-flan-t5-xl`.

With `openai/clip-vit-base-patch32`, $d_V = 768$. For FLMR, we use $N_{vt} = 32$ visual tokens per image representation and $d_L = 128$. For DPR, we use $N_{vt} = 6$ and $d_L = 768$ so that the number of parameters of vision mapping network is similar to that of FLMR:

$N_{vt} \times d_L \sim 128 \times 32$. The mapping network consists of two fully-connected layers with tanh activation. The output of last layer is reshaped into $N_{vt} \times d_L$ visual tokens. Other model parameters are: $l_q = 512$, $l_d = 512$. $N_{ROI} = 9$ unless otherwise specified.

We use 1 Nvidia A100 (80G) for all experiments. The optimizer is Adam [149]. In training the retrievers, we use learning rate 10^{-4}, batch size 30, gradient accumulation steps 2 for 10k steps (for both DPR and FLMR retrievers). When training RA-VQA-v2 (T5-large), we use learning rate 6×10^{-5}, batch size 2, gradient accumulation 16 for up to 20 epochs. We use a linearly-decaying scheduler to reduce learning rate from 6×10^{-5} to 0 after 20 epochs. We use LoRA [110] to train RA-VQA-v2 (BLIP2) with learning rate 10^{-4}, batch size 4, gradient accumulation steps 16 for up to 6k steps. LoRA is configured to use the default huggingface-PEFT setting: `r=8, lora_alpha=32, lora_dropout=0.1`.

The vision model is frozen throughout all experiments. In pre-training the mapping network, only the mapping network is trainable. When training the answer generator, the retriever is frozen.

We report the required GPU hours on 1 Nvidia A100 (80G): for vision-language alignment of retrieval models, approximately 4 GPU hours are needed. Training the FLMR retriever requires around 12 GPU hours (10k steps) including the time of running testing after training is complete. Training RA-VQA-v2 (BLIP 2) with LoRA requires around 12 GPU hours (6k steps) including the time of running validation per 1k steps. Training the RA-VQA-v2 (T5-large) requires around 12 GPU hours (3k steps) including the time of running validation every 500 steps.

A.3 Artifacts and License

We list the resources used and their License below:

(1) huggingface-transformers (Apache License 2.0) provides pre-trained model checkpoints for BLIP 2, DPR, T5, and their tokenizers: https://github.com/huggingface/transformers

(2) PLAID engine and ColBERTv2 checkpoints (MIT License): https://github.com/stanford-futuredata/ColBERT

(3) FAISS [138] (MIT License) is used to index document embeddings for fast retrieval with DPR: https://github.com/facebookresearch/faiss

(4) huggingface-PEFT (Apache License 2.0) for parameter-efficient LoRA fine-tuning: https://github.com/huggingface/peft

(5) The official RA-VQA implementation (GNU General Public License v3.0): https://github.com/LinWeizheDragon/Retrieval-Augmented-Visual-Question-Answering.

A.4 Retrieving Multi-modal Documents with FLMR

We additionally show that our proposed FLMR system can also be used to retrieve multi-modal documents. Since this is not the focus of the chapter, we present the investigation in this appendix.

Dataset. We select a subset from WIT [290], a knowledge corpus based on Wikipedia where the images associated with the documents are also present, to make an image-text corpus for retrieval. We adopt the same selection process as for the Wikipedia corpus introduced in Sec. 4. The dataset statistics is shown in Table A.2.

Multi-Modal Late Interaction. We upgrade the document embedding process to accommodate the document image. The documents in the knowledge base are represented by embeddings $\mathbf{D}$ which are obtained from the document content d and its associated image I_d:

$$\mathbf{D} = [\mathscr{F}_L(d), \mathscr{F}_M(\mathscr{F}_V(I_d))] \in \mathscr{R}^{l_D \times d_L}, \tag{A.1}$$

where $l_D = l_d + N_{vt}$, and l_d is the length of the document d.

We compute the relevance score between a question-image pair $\bar{\mathbf{q}} = (q, I)$ and a document $\bar{\mathbf{d}} = (d, I_d)$ as follows:

$$r(\bar{\mathbf{q}}, \bar{\mathbf{d}}) = r((q,I),(d,I_d)) = \sum_{i=1}^{l_Q} \max_{j=1}^{l_D} \mathbf{Q}_i \mathbf{D}_j^\top. \tag{A.2}$$

Discussion. Both query and document embeddings are multi-modal. Since the same image/text encoder is used to encode images I, I_d and texts q, d, respectively. Image-wise and text-wise relevance contribute to the final relevance score; After cross-modality alignment, the vision encoder $\mathscr{F}_M(\mathscr{F}_V(\cdot))$ should produce image embeddings close to the text embeddings produced by $F_L(\cdot)$ in the latent space if the image is relevant to the question, thereby taking the relevance between I, d and q, I_d into account during knowledge retrieval.

As shown in Table A.3, the retrieval scores see a slight improvement when document images are also considered (from text-only to multi-modal). This suggests that FLMR supports retrieving multi-modal documents.

However, we note that the gain of incorporating images is marginal. This is because WIT is a strongly text-driven knowledge base as the images are already captioned by human experts. The surrounding texts of images are already dense and informative, which can be searched by FLMR easily. By manual inspection, we also notice that it is very rare that OK-VQA questions seek a document that can only be found by its accompanying images. This also explains the marginal gain we have observed.

In conclusion, we show that FLMR can also be applied to retrieve multi-modal documents, although more challenging questions and better datasets are needed to fully exploit its potential. We leave this as future work.

Table A.3 FLMR performance when retrieving documents in WIT. Models suffixed by 'text/image-only' only encode document texts/images, while 'multi-modal' variants encode document images with vision encoders.

Model	PRRecall@5	PRRecall@10
DPR-text-only	68.24	77.13
DPR-image-only	46.29	57.70
DPR-multi-modal	68.78	77.90
FLMR-text-only	72.63	81.52
FLMR-image-only	45.75	57.92
FLMR-multi-modal	**73.65**	**81.89**

A.5 Effects of Retrieved Knowledge

Table A.4 Comparing Hit Success Rate of RA-VQA-v2 and RA-VQA.

	Hit Success Rate
RA-VQA-v2 (BLIP2)	9.38
RA-VQA (BLIP2)	7.86
RA-VQA-v2 (T5-large)	17.62
RA-VQA (T5-large)	15.01

It is important to understand the task performance that a base model has attained and the gains from knowledge retrieval. We use the evaluation metrics from RA-VQA (Sec. 3.4.2): the **Hit Success Ratio (HSR)** which counts questions that cannot be answered by the base VQA model alone and thus require external knowledge to answer.

$$HSR = \mathbb{1}\{\widehat{y} \in \mathscr{S} \wedge \widehat{y}_{NK} \notin \mathscr{S}\}, \tag{A.3}$$

where y_{NK} denotes the generated answer from a fine-tuned base model when no relevant knowledge is provided. HSR reflects the net value of incorporating external documents into answer generation. We can conclude from Table A.4 that RA-VQA-v2 steadily improves the HSR of RA-VQA by $\sim 2\%$, showing that the gains in VQA performance come from

improved knowledge retrieval. We also observe that T5-large, as an earlier language model, relies more heavily on retrieved knowledge (>15 HSR). This is because the base language model of BLIP 2, Flan-T5-XL, is significantly stronger and is able to answer more questions without the aid of external knowledge. This suggests that KB-VQA performance can be improved by either (1) applying stronger base VQA answer generation models, and (2) collecting knowledge documents of higher quality.

Table A.5 Performance improvements with increasing number of retrieved documents.

	K	5	10	20	50
DPR + T5-large	VQA Score	51.5	51.8	52.3	52.1
	Recall	83.08	89.77	94.05	97.25
FLMR + T5-large	VQA Score	54.9	55.3	55.4	55.4
	Recall	89.32	94.02	96.87	98.67

We also conduct experiments while increasing K in Table 5.1 and find that the system performance improves gradually until a saturation point. We notice that the saturation point of FLMR is at around $K = 10$ while that of DPR is at $K = 20$. This suggests that the useful documents are clustered around higher ranks in FLMR compared to DPR.

A.6 Computational Cost

We report the computational cost in this section.

	Train per 1000 steps	Indexing time
FLMR	1.2h	0.28h
w/o ROI	1h	0.25h
w/o ROI & VE	0.7h	0.24h
DPR	0.5h	0.2h

Table A.6 Training and indexing time for FLMR and DPR. Training batch size is 30. The corpus for counting the indexing time is the Google Search Corpus for OK-VQA (∼160k documents).

Though Late Interaction allows rich interactions at token level and outperforms DPR by a large margin, it also introduces additional latency in retrieval. As shown by Table A.6, the training time of FLMR is increased from 0.5h to 0.7h when late interaction is introduced. This latency increase comes from the more complicated token-level loss. When Vision

Encoder (VE) and ROI (Region of Interest) are added, the time cost is increased to 1h and 1.2h respectively due to the additional trainable parameters of the mapping network (the parameters of the visual encoder are not used in training since the visual features can be easily pre-computed and cached). However, the indexing time does not increase significantly when VE and ROI are added to the FLMR retriever. We note that the FLMR spends slightly more time to build the search index when compared to DPR because an extra clustering step by PLAID [263] is required to conduct fast retrieval.

Retriever	Generator	Training (iterations/sec)	Inference (iterations/sec)
FLMR	T5-large	1.16	1.11
DPR	T5-large	1.73	1.67
FLMR	BLIP 2	1.24	0.98
FLMR (w/o ROI & VE)	BLIP 2	1.43	1.00
DPR	BLIP 2	2.14	1.30

Table A.7 Training and inference time of the whole system. Please note that passages are dynamically retrieved, and thus the training and inference time already takes the retrieval latency into account. Batch size is set to 1 for both training and inference time. *w/o ROI & VE* means removing the vision encoder in FLMR.

When FLMR is integrated into the full VQA pipeline (we take the BLIP 2 version for example), it reduces the training speed from 2.14 iterations/sec to 1.24 iterations/sec (42%) since the retrieval process is run on the fly. However, in retrieval, the inference speed is only reduced from 1.3 iterations/sec to $\sim$1.0 iterations/sec, which is still affordable when considering the performance boost. The major computational cost remains that of training the answer generator with a great number of parameters.

Appendix B

Appendix for Chapter 6

B.1 Datasets details

This section outlines the preprocessing methods used to convert various datasets into formats suitable for retrieval tasks. Subsequent subsections detail the specific preprocessing steps for each dataset. The M2KR dataset is available at Huggingface Hub: https://huggingface.co/datasets/BByrneLab/multi_task_multi_modal_knowledge_retrieval_benchmark_M2KR.

B.1.1 I2T Retrieval

WIT

WIT [290] is a corpus based on Wikipedia with image-text pairs, where the text is the Wikipedia passage associated with the image. To enhance data quality, we exclusively select image-text pairs where the images are the main/title images of their respective Wikipedia documents, and we limit our scope to English-language documents.

Our training set, comprising 2.8 million examples, is sourced from the original WIT training set. 20,102 and 5,120 examples from the original WIT validation set are selected to build the validation set and test set in our M2KR benchmark, respectively. The test corpus includes all documents from the original WIT validation and test sets. This setting ensures that there is no overlap between different sets.

Each image-document pair is paired with a randomly selected instruction from our set of templates. The task is to retrieve the correct document from the test corpus, given the image and instruction.

IGLUE

The IGLUE English retrieval test set [24], which is a subset of the WIT test set and has an established benchmark for image-to-text retrieval, is included to enable comparison with models in previous literature. Following Bugliarello et al. [24], the test set contains 685 unique images and 1,000 Wikipedia passages. The task is similar to WIT: using the image and the instruction to retrieve the corresponding Wikipedia passage.

Instruction templates for WIT and IGLUE:

- <Image> Identify the document that is connected to this image.
- <Image> Provide information about the document linked to this image.
- <Image> Please describe the document that corresponds to this image.
- <Image> What is the document that this image is related to?
- <Image> Could you elucidate the document associated with this image?
- <Image> Describe the document that accompanies this image.
- <Image> Please give information on the document that goes with this image.
- <Image> What document is represented by this image?
- <Image> Identify the document that this image pertains to.

KVQA

KVQA [275] is a dataset containing a rich collection of entities representing famous individuals. The KVQA task, initially designed as a KB-VQA task, has been re-purposed into an I2T task for our modelling purposes. This adaptation is based on our findings that using images as queries alone suffices to retrieve the documents containing the correct identities. In our context, where the primary focus is on document retrieval, the original questions are unnecessary. Our reformulated task for KVQA is to retrieve the details of famous people like gender, nationality, birthplace, and employment history based solely on their images. The training set is downsampled from the KVQA original training set by removing repeated examples of the same famous individuals. We transformed the structured entities such as gender and nationality into passages. For example, "nationality: America; date of birth: dd/mm/yyyy; ..." is serialised as "nationality is America, date of birth is dd/mm/yyyy, ...".

The training corpus is composed of all the documents that appear in the original KVQA training set. For the validation/test set, we selected a subset of 13,365/5,120 samples from the original KVQA validation set. Correspondingly, the test corpus encompasses all documents found in the original KVQA validation set.

The instruction we use for KVQA is: <Image> Provide a brief description of the image and the relevant details of the person in the image.

CC3M

CC3M [278] is a dataset consisting of a vast collection of image-caption pairs. Instead of utilising the entire dataset comprising 3 million pairs, we adopt the downsampling methodology as delineated in LLaVA's work [198], resulting in a reduced dataset of approximately 595K.

We reformulate the image-caption pairs into image-to-text retrieval tasks in our pre-training. To construct the training corpus, we treat each caption as an individual document linked to its corresponding image. The task then involves retrieving the most relevant caption for a given image, guided by a set of randomly selected instructions. Since CC3M is originally an image captioning task, we do not validate or test our retriever on CC3M.

Instruction templates for CC3M

- <Image> Describe the image concisely.
- <Image> Provide a brief description of the given image.
- <Image> Offer a succinct explanation of the picture presented.
- <Image> Summarize the visual content of the image.
- <Image> Give a short and clear explanation of the subsequent image.
- <Image> Share a concise interpretation of the image provided.
- <Image> Present a compact description of the photo's key features.
- <Image> Relay a brief, clear account of the picture shown.
- <Image> Render a clear and concise summary of the photo.
- <Image> Write a terse but informative summary of the picture.
- <Image> Create a compact narrative representing the image presented.

B.1.2 Q2T Retrieval

MSMARCO

MSMARCO [16] stands for Microsoft Machine Reading Comprehension dataset. It is a text-only dataset with around 1 million questions and 8 million passages. At stage 0, we train according to ColBERT-v1 by Khattab and Zaharia [146]. For later stages, we downsample the dataset to 400K questions to balance between the multi-modal tasks and uni-modal tasks. For the training corpus, we still use the full 8 million passages. For testing, we select 6,980 and 5,120 samples from the original MSMARCO validation set and sample 400K passages to retrieve from and ensure the subset contains all ground-truth passages.

Instruction templates for MSMARCO:

- <Blank image> Retrieve the document that answers this question. <Questions>
- <Blank image> Find the document that is most relevant to the question. <Questions>

- <Blank image> Obtain the document that resolves this query. <Questions>
- <Blank image> Acquire the document that elucidates this question. <Questions>
- <Blank image> Choose the document most relevant to the query. <Questions>
- <Blank image> Identify the document most applicable to the question. <Questions>
- <Blank image> Extract the document that answers this query. <Questions>
- <Blank image> Locate the document that addresses the query.<Questions>

B.1.3 IQ2T Retrieval

LLaVA

The LLaVA instruction following dataset contains GPT-3.5 generated high-quality conversation about an image between a human and an AI assistant. There are around 150K rounds of conversations. We took each conversation (each question from the human and the answer from the AI assistant) as a separate sample. This results in a total of 356K samples. Since there are no original validation or test sets associated with the LLaVA, we manually split the sample pool into 351K training examples and 5,120 test examples.

The task is reformulated to an Image&Question to Text retrieval task. The training corpus and test corpus each contain the associated answers as passages to be retrieved by the image and question pairs. We use two types of instruction templates depending on the preciseness of the question:

- <Image> Provide a brief description of the image along with the following question: <Question>
- <Image> Provide a concise explanation of the image along with the following question: <Question>

OVEN

OVEN is a dataset targeting open-domain visual entity recognition. The dataset consists of two splits: entity set and query set. The entity set is derived from image classification datasets such as INaturalist2017 [60], Food-101 [22], Cars196 [151] and Google Landmarks Dataset v2 [321]. The query set is derived from VQA datasets such as VQAv2 [94] and OK-VQA [273]. To avoid overlapping with our other KB-VQA datasets, we only use the entity set of OVEN. The entity set contains about 10K unique entities.

The original entity set contains about 5 million question-image pairs. However, the questions are highly duplicated in the original OVEN dataset. We downsample the dataset by removing repeated questions corresponding to the same entity. This reduces duplications while maximizing the diversity of the questions and coverage of entities. After the filtering,

we keep 339K training samples. For validation and testing, we select 20,000 and 5,120 examples from the original OVEN Entity validation set. The original test set is not used in M2KR due to the lack of annotation.

The original task is to link the image to a specific Wikipedia Entity given a question. To formulate the task as a retrieval problem, for each entity, we use its associated Wikipedia passage as the document to retrieve. The query side of this retrieval task contains the image and its question with the inclusion of a randomly sampled instruction. Given this query, the task is to obtain the relevant Wikipedia passage. The training corpus contains about 10K passages, while the test corpus contains about 3.2K passages that cover all entities in OVEN's original training set and validation set respectively.

E-VQA, Infoseek and OK-VQA

E-VQA, Infoseek, and OK-VQA are Knowledge-based VQA (KB-VQA) datasets. For each given image and question (with instruction), the task is to retrieve the corresponding knowledge passage.

For **E-VQA** [218], the original training set contains around 1 million samples. However, it includes duplicated questions and answers referring to the same Wikipedia Entity with different query images. We filter duplicated questions that pertain to the same Wikipedia Entity. To align with the original evaluation setting of E-VQA, we further excluded samples that necessitate multiple knowledge bases, reducing the count to 167K training samples. To be consistent with the original E-VQA paper, our validation and testing sets exclusively include questions that can be answered using a single knowledge passage. These sets contain 9,852 and 3,750 samples, respectively. We use the WikiWeb2M [26] as the knowledge source. For the training and test passage corpus, we keep all the passages that appear in the original E-VQA to align with the official E-VQA's setting for retrieval.

For **OK-VQA**, we use the original training and test set. Following Lin et al. [188], we prepare a knowledge corpus with Wikipedia documents based on pseudo-relevance. The training and test passage corpus both contain all passages in the knowledge corpus.

For **Infoseek**, following the preprocessing steps described by Chen et al. [44], we use Wikipedia documents as knowledge sources and remove examples whose answers can not be found in the ground-truth documents. We randomly selected 100K examples from the training set for training and 4,708 examples from the validation set for testing (the annotation of the original test set has yet been released). The downsampling is motivated by our observation that many questions are repeated and the number of unique documents associated with the whole dataset is only about 40K. We downsampled the dataset such that the model won't overfit severely to Infoseek passages.

Note that the aforementioned downsampling procedure for the test set is only used for constructing the M2KR benchmark. For downstream VQA evaluation, we use the same test set that existed in previous literature to ensure a fair comparison.

Instruction templates for OVEN, Infoseek, E-VQA, and OK-VQA

• <Image> Using the provided image, obtain documents that address the subsequent question: <Question>

• <Image> Retrieve documents that provide an answer to the question alongside the image: <Question>

• <Image> Extract documents linked to the question provided in conjunction with the image: <Question>

• <Image> Utilizing the given image, obtain documents that respond to the following question: <Question>

• <Image> Using the given image, access documents that provide insights into the following question: <Question>

• <Image> Obtain documents that correspond to the inquiry alongside the provided image: <Question>

• <Image> With the provided image, gather documents that offer a solution to the question: <Question>

• <Image> Utilizing the given image, obtain documents that respond to the following question: <Question>

B.2 Implementation Details

B.2.1 Breakdown of Data Used in Training

The detailed breakdown of the data used in different phases is presented in Table B.1. In Stage 3 Full-scale Fine-tuning, the different sub-tasks in the M2KR dataset are downsampled or duplicated to balance the dataset proportions during training. We observed that without adjusting the data proportions during training, the model's training losses on certain datasets like WIT, Infoseek, and OVEN decrease much faster than on others once all parameters become trainable. This goes against our goal of training a multi-tasking system. Adjusting the data proportions is crucial to ensure a more consistent learning process across different tasks.

	Stage 1	Stage 2	Stage 3
WIT	2.8M	-	140K
IGLUE	-	-	-
KVQA	65K	-	6.5K
CC3M	595K	-	29.8K
MSMARCO	400K	-	40K
OVEN	339K	-	33.9K
LLaVA	351K	-	35.1K
OK-VQA	9K	-	90K (repeat 10 times)
Infoseek	100K	-	50K
E-VQA	167K	167K	167K

Table B.1 The dataset sizes are adjusted in Stage 3 in practice.

B.2.2 Detailed Hyperparameters

We use the Adam optimizer [149] with a fixed learning rate of 10^{-4} for the mapping structure and 10^{-5} for the rest parameters in all experiments in all training stages. 4 Nvidia A100 GPUs were used with data parallel in all experiments.

Stage 0: Training was run up to 300k steps. The batch size is 8 and the gradient accumulation step is 8. The number of negative examples is 1. The validation ran per 10k steps. The checkpoint was taken at the best Recall@50 on the original MSMARCO validation set, following Khattab and Zaharia [146]. The total training time is approximately 1.5 days per model.

Stage 1: Training was run up to 220k. The batch size is 8 and the gradient accumulation step is 8. The number of negative examples is 4. The validation interval is 10k steps. The checkpoint was taken at the best Recall@10 on the validation set of WIT in M2KR. The total training time is approximately 5 days per model.

Stage 2: The intermediate pre-training was run for 12k steps for all experiments. The batch size is 8 and the gradient accumulation step is 8. The number of negative examples is 4. The total training time is approximately 2 days per model.

Stage 3: Training was run for 50k for all experiments. Training was early-stopped if the performance on WIT or E-VQA decreases for 3 consecutive validation runs. Validation was run per 10k steps. The batch size is 8 and the gradient accumulation step is 8. The number of negative examples is 4. The total training time is approximately 2 days per model.

Single-task Downstream Fine-tuning: The batch size is 8 and the gradient accumulation step is 8. The number of negative examples is 4. For reference, in our experiments, the downstream fine-tuning took 20k, 5k, 1k, 15k, 2.5k steps to achieve the best performance for

WIT, OVEN, Infoseek, E-VQA, OK-VQA respectively (for the ViT-G + Base-v2 PreFLMR). The total training time is approximately 1 days per model per task.

VQA Fine-tuning: We used BLIP2-T5XL as the answer generator as in RA-VQA-v2 [188]. The retriever was frozen during training and inference. The batch size is 1 and the gradient accumulation step is 16. For each question in a training batch, top-5 relevant documents were pre-extracted using the retriever, and 3 out of 5 were randomly selected. These 3 documents were concatenated to the question and sent to the answer generator for one forward pass individually. This setting is to enable training with top-5 documents given limited GPU memory. The total training time is approximately 2 days per model.

Every model reported in this paper was reproduced once to make sure the training is reproducible. The best result is reported since the model with the best result will be released to the community. There is not much difference in the two runs. The absolute difference is less than 0.2 Recall score in most datasets (except that PreFLMR_ViT-B_Base-v2 has a -0.4 difference on Infoseek).

B.2.3 Large-v1 Training

In our experiments, we found that training Large-v1 during Stage 2/3 was not steady. First, the loss decreased faster than in other systems, like Base-v1, even though Large-v1 had worse system performance. This happened because Large-v1's bigger model capacity made it more prone to overfitting.

Next, the loss suddenly shot up, causing the training to collapse, despite using the same data and strategy as Base-v1. We tried different hyperparameters, like lowering the learning rate to $1e-6$, $3e-6$, but the model still collapsed.

Finally, when we used LoRA [110] with Large-v1 during training, it helped stabilise the process. The LoRA hyperparameters used were: $r = 16$, $\alpha = 32$, and a dropout rate of 0.05.

B.2.4 Model Design in Detail

Similar to FLMR, PreFLMR consists of three components: a vision model $\mathscr{F}_V$, a mapping structure $\mathscr{F}_M$, and a language model $\mathscr{F}_L$.

Feature Extraction. The textual query q consists of an instruction and (optionally) a question (e.g., "Utilise the given image to procure documents addressing the following query: [Question]"). We use a language model with hidden size d_L to obtain embeddings for all N_q tokens which are concatenated into matrix $\mathbf{Q}_q$:

$$\mathbf{Q}_q = \mathscr{F}_L(q) \in \mathscr{R}^{N_q \times d_L}. \tag{B.1}$$

Like FLMR, a vision model $\mathscr{F}_V$ encodes the input image I, extracting the '[CLS]' token embeddings from the last layer. PreFLMR additionally uses the patch embeddings from the penultimate layer of ViT for more complete representation.

$$\mathbf{Q}_{I,[CLS]} = \mathscr{F}_V(I) \in \mathscr{R}^{1 \times d_V}; \tag{B.2}$$

$$\mathbf{Q}_{I,PATCH} = \mathscr{F}_{V,-2}(I) \in \mathscr{R}^{N_V \times d_V}. \tag{B.3}$$

The mapping structure $\mathscr{F}_M$ comprises two components: a 2-layer MLP $\mathscr{F}_M^{MLP}$ and a Transformer block $\mathscr{F}_M^{TR}$.

Following the FLMR model, a 2-layer Multi-Layer Perceptron (MLP) $\mathscr{F}_M^{MLP}$ is utilised to convert the initial token embeddings into visual token embeddings with a length of N_{vt} and a hidden size d_h:[1]

$$\mathbf{Q}_I^{MLP} = \mathscr{F}_M^{MLP}(\mathbf{Q}_{I,[CLS]}) \in \mathscr{R}^{N_{vt} \times d_h}. \tag{B.4}$$

Moreover, an additional Transformer module $\mathscr{F}_M^{TR}$ is introduced to manage all patch embeddings. It is a stack of N_{TR} transformer layers with a hidden size d_L, followed by a simple MLP layer at the end. This module leverages cross-attention with the text query $\mathbf{Q}_q$, enabling query-aware image feature mapping.

$$\mathbf{Q}_I^{TR} = \mathscr{F}_M^{TR}(\mathscr{F}_v(\mathbf{Q}_{I,PATCH}), \mathbf{Q}_q) \in \mathscr{R}^{N_V \times d_h}. \tag{B.5}$$

Here, $\mathscr{F}v$ represents a 1-layer MLP that adapts the dimension from d_V to d_L, which is subsequently transformed to d_h by the linear MLP layer of $\mathscr{F}_M^{TR}$. The resultant features from these processes are concatenated to formulate the query embeddings:

$$\mathbf{Q} = \left[\mathbf{Q}_q | \mathbf{Q}_I^{MLP} | \mathbf{Q}_I^{TR}\right] \in \mathscr{R}^{(N_{vt}+N_V+N_q) \times d_h}. \tag{B.6}$$

Furthermore, the document representations in the knowledge base are denoted by $\mathbf{D}$, derived from the document content d with length l_D:

$$\mathbf{D} = \mathscr{F}_l(\mathscr{F}_L(d)) \in \mathscr{R}^{l_D \times d_h}, \tag{B.7}$$

[1]Transformation sequence: $\mathscr{R}^{d_V} \to \mathscr{R}^{N_{vt} d_h/2} \to \mathscr{R}^{N_{vt} d_h}$, subsequently reshaped into $\mathscr{R}^{N_{vt} \times d_h}$.

where $\mathscr{F}_l$ signifies a straightforward MLP layer tasked with mapping d_L to d_h, thereby aligning the dimensionality with the query embeddings.

Multi-Modal Late Interaction. The relevance score between a question-image pair $\bar{\mathbf{q}} = (q, I)$ and a document d is calculated using a late-interaction paradigm:

$$r(\bar{\mathbf{q}}, d) = r((q, I), d) = \sum_{i=1}^{l_Q} \max_{j=1}^{l_D} \mathbf{Q}_i \mathbf{D}_j^\top, \tag{B.8}$$

where $l_Q = N_{vt} + N_V + N_q$. For each token in the query, the system aggregates the maximum relevance score across all tokens in the document.

Training and Inference. For model training, documents d^* corresponding to a query q are considered gold (positive) samples. We incorporate random negative sampling from the corpus.[2] Additionally, we adopt in-batch negative sampling as suggested by Karpukhin et al. [143], treating all non-corresponding documents in a batch as negatives for q, denoted as $\mathscr{N}(q)$. The model is trained using a contrastive loss across the dataset $\mathscr{D}$:

$$\mathscr{L} = -\sum_{(q,d^*)\in\mathscr{D}} \log \frac{\exp\left(r(q,d^*)\right)}{\exp\left(r(q,d^*)\right) + \sum_{z\in\mathscr{N}(q)} \exp\left(r(q,z)\right)}. \tag{B.9}$$

Post-training, all documents are indexed through PLAID [263] for efficient late-interaction retrieval. For detailed evaluation of retrieval efficiency, we refer readers to Sec. A.6.

B.3 Ablation Study on Pre-training Stages

Model	WIT	IGLUE	KVQA	MM	OVEN	LLaVA	Infoseek	E-VQA	OK-VQA
PreFLMR_ViT-B_Base-v1	41.7	57.3	28.6	79.5	46.3	67.2	48.8	67.9	66.1
w/o Stage 0	25.5	28.8	21.0	56.5	33.9	55.0	42.5	51.8	64.5
w/o Stage 1	38.2	54.9	26.6	78.0	45.5	62.8	44.6	61.9	65.5
w/o Stage 2	41.2	56.8	26.5	78.2	43.7	65.0	47.0	57.3	65.1

Table B.2 Retrieval performance when disabling pre-training stages. Removal of any stage deteriorated the performance.

We present the ablation study for the four pre-training stages in Table B.2. To ensure consistent comparison, these ablated versions underwent the same number of training steps as PreFLMR_ViT-B_Base-v2. The results clearly indicate that the removal of any stage deteriorates performance. Specifically, disabling Stage 0 (i.e. using untrained text encoder)

[2] In multi-dataset scenarios, negative samples are selected from the same corpus as d^*.

leads to the most significant performance decline because the text encoder is not pre-trained on late-interaction, resulting in a diminished ability to capture fine-grained relevance within the same computational budget. Note that removing Stage 0 leads to collapsed performance on Stage 1, where the text encoder is frozen. Furthermore, removing Stage 2 notably affects the performance on E-VQA more than on other KB-VQA datasets, highlighting the challenge posed by E-VQA and the necessity of intermediate pre-training.

B.4 V-Entropy-based Analysis of Intermediate Pre-training

V-Entropy [335], $H_{\mathscr{V}}(Y|X)$, is the minimal Negative Log-Likelihood (NLL) achievable by the probabilistic predictor $f(Y|X)$ under the predicative family $\mathscr{V}$. A predicative family can be viewed as the set of reachable models under a certain model architecture and training budgets.

We define Mutual Information $I_{\mathscr{V}[N_f]}(D_1 \to D_2)$ between datasets D_1 and D_2 in Eq.B.10. We define $H_{\mathscr{V}[N_f]}(D_2)$ as the minimal achieved NLL loss on the validation set of dataset D_2 after N_f training steps on D_2. $\mathscr{V}[N_f, D_1, N_t]$ denotes the set of reachable models after N_f fine-tuning steps on D_2 starting from a checkpoint that has been trained on dataset D_1 for N_t steps. This is V-Entropy with additional predictive family specification.

$$I_{\mathscr{V}[N_f]}(D_1 \to D_2) = H_{\mathscr{V}[N_f]}(D_2) - H_{\mathscr{V}[N_f, D_1, N_t]}(D_2). \tag{B.10}$$

Intuitively, D_1 has high mutual information with D_2 if models initialised from D_1 checkpoints attain much lower NLL loss compared to models initialised without training on D_1. N_f and N_t set the computation constraints for training on D_2 and D_1, respectively. In our experiment, $\mathscr{V}$ is the PreFLMR architecture, D_1 is the E-VQA dataset and D_2 is the training set of M2KR. N_f corresponds to N_{inter} in Sec. 6.5.6, which is the intermediate training steps on the E-VQA dataset. In the analysis, we set N_f to 5,000 and sweep N_t from 0 to 25,000 in intervals of 5,000.

We refer readers to Xu et al. [335] for detailed properties of V-Entropy and emphasise that $I_{\mathscr{V}[N_f]}(D_1 \to D_2)$ is an empirical value we define to estimate mutual information between datasets. It is different from the V-Information defined in Xu et al. [335] which estimates the mutual information between model input and output.

B.5 Qualitative Analysis for OK-VQA and E-VQA

In this section, we compare examples from the OK-VQA and E-VQA datasets to highlight their differences. To avoid cherry-picking, we use examples from its official website[3] for OK-VQA. Similarly, we use the examples included in the paper for E-VQA. Table B.3 presents three examples from each dataset.

The OK-VQA examples typically require common sense knowledge, like 'people attend church on Sundays' or 'firetrucks use fire hydrants'. State-of-the-art Large Language Models (LLMs) often have this common sense knowledge inherently built-in, making additional knowledge retrieval less impactful for OK-VQA tasks.

In contrast, E-VQA examples demand more specialised, expert-level knowledge, necessitating an effective knowledge retrieval system. For instance, correctly answering a question about 'Acacia paradoxa' requires first retrieving the relevant document providing specific information about this plant species. Enhancing the knowledge retrieval system to source accurate documents is crucial for improving performance on the E-VQA dataset.

B.6 Artifacts and License

We list the resources used and their License below:

(1) huggingface-transformers (Apache License 2.0) provides pre-trained model checkpoints for BLIP 2, DPR, and their tokenizers: https://github.com/huggingface/transformers.

(2) FAISS [138] (MIT License) is used to index document embeddings for fast retrieval with DPR: https://github.com/facebookresearch/faiss.

(3) huggingface-PEFT (Apache License 2.0) for parameter-efficient LoRA fine-tuning: https://github.com/huggingface/peft.

(4) PLAID and ColBERTv2 (MIT License): https://github.com/stanford-futuredata/ColBERT.

(5) RA-VQA-v2 official repository with training and testing codes (GNU General Public License v3.0): https://github.com/LinWeizheDragon/Retrieval-Augmented-Visual-Question-Answering.

(6) Datasets used in building the M2KR benchmark:

- WIT (Creative Commons Attribution-ShareAlike 3.0 Unported https://github.com/google-research-datasets/wit/blob/main/LICENSE);
- MSMARCO (non-commercial research purposes only https://microsoft.github.io/msmarco/);
- CC3M (Free for any purposes https://github.com/google-research-datasets/conceptual-captions);

[3] https://okvqa.allenai.org/

OK-VQA		
		
Q: What days might I most commonly go to this building? A: Sunday	Q: What sort of vehicle uses this item? A: firetruck	Q: Is this photo from the 50's or the 90's? A: 50's
E-VQA		
		
Q: How many feet tall does this tree grow to? A: 7 to 13	Q: How many eggs does this reptile typically lay? A: 3-6	Q: Who founded this monastery? A: Prince Constantin Brâncoveanu

Table B.3 Demonstrative examples from OK-VQA and E-VQA. Questions in E-VQA require more domain knowledge to answer generally.

- LLaVA, the image of LLaVA is a subset of CC3M. It should inherit the license of CC3M. The conversation data follows policy of OpenAI: https://openai.com/policies/terms-of-use;
- IGLUE (MIT license https://github.com/e-bug/iglue/blob/main/LICENSE);
- KVQA (No specific license is mentioned https://malllabiisc.github.io/resources/kvqa/);
- OVEN (Apache-2.0 license https://github.com/open-vision-language/oven/blob/main/LICENSE);
- E-VQA (no specific license mentioned https://github.com/google-research/google-research/tree/master/encyclopedic_vqa);
- Infoseek (Apache License 2.0 https://github.com/open-vision-language/infoseek/blob/main/LICENSE);
- OK-VQA (Copyright (c) 2021, Chen Qu and Center for Intelligent Information Retrieval, University of Massachusetts, Amherst. https://github.com/prdwb/okvqa-release/blob/main/LICENSE).

In particular, we emphasise that no changes are made to the original data of all the datasets used in our work. Our released models and artifacts should only be used for non-commercial purposes. By using the pre-trained models, users agree to respect the terms and conditions of the datasets used in pre-training.

B.7 PreFLMR model performance radar chart on M2KR tasks

Fig. B.1 demonstrates the performance of PreFLMR with a radar plot. The best and worst numbers of each task are annotated.

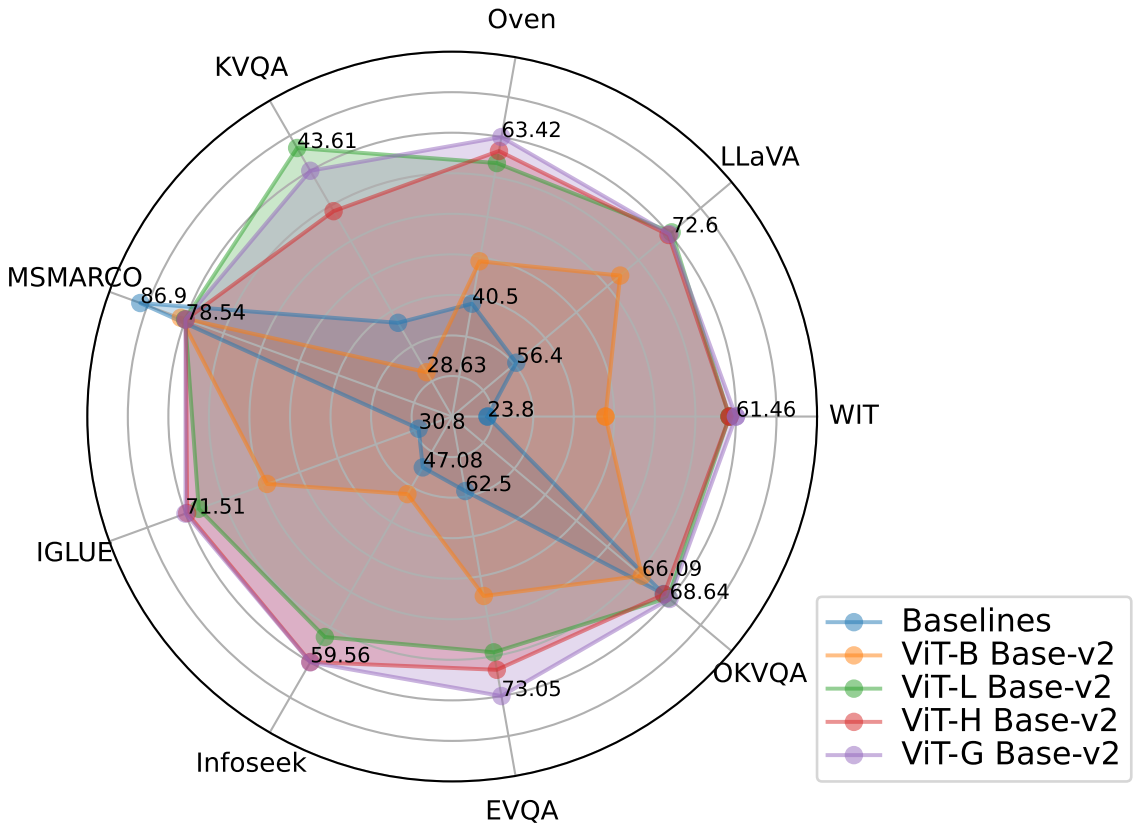


Fig. B.1 PreFLMR achieves strong performance on the M2KR benchmark. The scale of the plot is adjusted for better visualisation. The best and worst numbers of each task are annotated.

Appendix C

Appendix for Chapter 7

C.1 Implementation Details

C.1.1 ITR Retriever Configuration

For the ITR retrieval component, i.e., to obtain question encoder (E_Q) and item encoder (E_T) in Algorithm 1, we fine-tune DPR on WikiSQL to retrieve the relevant table items for a question. We initialise E_Q and E_T using DPR weights released via `huggingface`, i.e., `facebook/dpr-question_encoder-single-nq-base` and `facebook/dpr-ctx_encoder-single-nq-base`, respectively. Before encoding via E_T, we linearise table items in a naive way with additional special tokens interleaving table cell values. For example, to encode row r_3 in Fig. 7.1(a) we use: *<HEADER> rank <HEADER_SEP> mountain peak <HEADER_SEP> mountain range <HEADER_SEP> elevation <HEADER_END> <ROW> 2 <ROW_SEP> red slate mountain <ROW_SEP> sierra nevada <ROW_SEP> 13 , 149 ft <ROW_END>*.

We obtain annotations for *gold cells* by assessing the SQL query associated with each question-table-answer triple (q, T, a) in WikiSQL. For example, we can evaluate the following SQL query annotation "`SELECT Header1 FROM table WHERE Another Header = Some Entity`" to obtain cells that are selected as answers. In case of other aggregation functions beyond cell selection (e.g. COUNT, SUM, AVG), gold cells are those selected as input to aggregation functions. Thus, the table items containing any of these gold cells are gathered into positive items $I^+(q,T)$ while the remaining negative items are in $I^-(q,T)$.

In training ITR encoders, we leverage a contrastive loss to increase the output similarities between the question embeddings $E_Q(q)$ and the embeddings of positive items $E_T(i)$ ($i \in I^+(q,T)$). In contrast, the similarities with the embeddings of negative items are reduced.

Parameter	Value
Negative samples	4 (per positive sample)
Total GPUs	8
Learning rate	0.0001
Optimizer	Adam
Batch size	1 (per device)
Grad. accum. steps	4
Training steps	3800 (ITR_{col}) 6800 (ITR_{row}) 11 600 (ITR_{mix})

Table C.1 Best hyperparameters chosen for ITR retriever on the WikiSQL dataset.

Formally, the embeddings of questions and items are:

$$\mathbf{e_q} = E_Q(q) \in \mathbf{R}^d; \mathbf{e_i} = E_T(i) \in \mathbf{R}^d, \tag{C.1}$$

where $i \in I \mid I = \texttt{Items}(T)$ and d is the hidden size. We use inner dot product as our similarity function:

$$r(q,i) = \mathbf{e_q}\mathbf{e_i}^\top. \tag{C.2}$$

For each question, one positive item i^* and a few negative items are randomly sampled from $I^+(q,T)$ and $I^-(q,T)$ respectively. The training loss is therefore:

$$-\sum_{(q,I)} \log \frac{\exp\left(r(q,i^*)\right)}{\exp\left(r(q,i^*)\right) + \sum_{i \in I^-(q,T)} \exp\left(r(q,i)\right)}. \tag{C.3}$$

In Table C.1 we report other hyperparameters for the ITR retrieval component, chosen based on the Dev set of WikiSQL. We recall that we use the same checkpoint trained on WikiSQL also for WikiTQ in TableQA task. We train ITR retrieval component on an V100 machine, and the total training time for our main ITR variant is about 380 minutes.

C.1.2 Training with ITR Configuration

We use only TaPEx as a baseline for ITR at training time. We initialise TableQA models with the released checkpoint from `huggingface` for TaPEx pre-training, i.e., `microsoft/tapex-large`. As previously shown in Table 7.6, we notice a slight difference on the performance of the released TaPEx checkpoints via `huggingface` and the in-house fine-tuned TaPEx. Due to this, we report the hyperparameters we use to fine-tune TaPEx on WikiSQL and

Parameter	Value (WikiSQL)	Value (WikiTQ)
Warmup steps	1000	0
Epochs	10	40
Learning rate	0.00003	0.00002
LR decay	Linear	
Optimizer	AdamW	
Total GPUs	8	
Batch size	1 (per device)	
Grad. accum. steps	4	
Weight decay	0.01	
Label smoothing	0.1	

Table C.2 Best hyperparameters chosen for the in-house and ITR-enhanced TaPEx for WikiSQL and WikiTQ datasets.

Baseline	Checkpoints
	WikiSQL
TaPEx	`microsoft/tapex-large-finetuned-wikisql`
TaPas	`google/tapas-large-finetuned-wikisql-supervised`
	WikiTQ
TaPEx	`microsoft/tapex-large-finetuned-wtq`
TaPas	`google/tapas-large-finetuned-wtq`
OmniTab	`neulab/omnitab-large-finetuned-wtq`

Table C.3 Checkpoints released via the `huggingface` library for TaPEx, TaPas and OmniTab, that we use as baselines for inference only experiments with ITR.

WikiTQ datasets in Table C.2. We choose the best hyperparameters based on the performance on Dev set of each benchmark.

C.1.3 Inference with ITR Models

In Table C.3 we report the model checkpoints from `huggingface` that we used as baseline when applying ITR at inference time only. As mentioned in Sec. 7.5, there are some differences between the performance obtained when evaluating the `huggingface` implementation of the baselines and the performance reported in each separate paper, mainly due to data processing and evaluation scripts. For example, OmniTab official repository was firstly based on that of TaBERT, which is based on encoder-only architecture. The authors have adjusted the code to an encoder-decoder architecture, however maintaining the tokenizer of TaBERT rather than TaPEx.This detail has not been transferred to the `huggingface` implementation. In addition, after the release of the original TaPas [106], authors have

implemented different variants on the same repository, including the preprocessing of the data and/or evaluation scripts. For example, Herzig et al. [106] report that they drop examples if certain conditions are not satisfied, such as there is no scalar answer and the denotation cannot be found in the table. It is not clear if this decision continues to be true for the subsequent developments. Therefore, this does not allow a straightforward assessment of ITR contribution. To this end, we unify the implementations in a single evaluation framework, using the dataset splits, checkpoints and evaluation methods made available in the `huggingface` library for all the baselines.

C.2 Column and Row Order Effect

Despite the order of items returned by ITR, after creating and choosing the sub-tables, we rearrange their columns and rows in the same order as that in the original table. We rely on the order of the original training data, which can have its own biases in data creation. In addition, we observe that:

1. Exposing the most relevant items first at training time, in which case we also have access to gold items, leads to quick model overfitting. The model can be strongly biased to choose cells that appear early in the linearised sequence, which is not desired for training a generalisable and robust TableQA model.

2. Baseline models have been trained without a strong bias on the column/row order, i.e., not enforcing that most relevant items are shown first. We show several experiments in which we apply ITR at inference time only. As such, introducing an ordering bias at inference time only decreases the performance.

Furthermore, to investigate whether the positioning of gold answers in the dataset can bias the trained model, we shuffle sub-table rows and columns to make the gold answers appear equally possible in any position of the input table. Results in Table 7.5 showed that shuffling at training time slightly increases the robustness of the model by 0.2-0.5 denotation accuracy points in WikiSQL Test and Dev sets respectively. Interestingly, shuffling has a bigger impact in the extreme scenario (see Sec. 7.5.2) increasing the $\text{Overflow}_{ext.}$ by 2 denotation accuracy points. In the literature different strategies have been employed in the model design to avoid positional biases. For example, TableFormer [339] disposed of positional embedding to make all token positions homogeneous. However, such modifications of the baselines are out of the scope of our work, in which we show the contribution of ITR on the current settings of each baseline.

N	WikiSQL Dev	WikiSQL Test
20	91.24	91.34
15	91.22	91.30
10	91.25	91.35
5	91.23	91.30
4	91.18	91.27
3	91.14	91.21
2	91.08	91.11
1	91.03	90.97
0	88.4	87.7

Table C.4 DA of ITR → TaPEx for varying values of *N*. Underlined values denote the performance at our chosen *N* for the best model. N=0 indicates the baseline, i.e., using the full table.

C.3 Multiple Sub-tables Effect

For our main experiments we use $N > 1$ sub-tables at inference time for generation baseline systems, i.e., TaPEx and OmniTab. In particular, we use $N = 10$ for our main ITR variant and ITR_{ngram}, while for the column and row only ITR variants, we set $N = 5$ and $N = 10$. In Sec.7.5 (and Fig. 7.3), we showed that ITR retrieval performance converges after $K > 5$ for columns and $K > 10$ for rows, which justifies the values selected in TableQA for N for each variant.

In Sec. 7.6 we showed the marginal impact of N=1 *versus* N=10. For completeness, in Table C.4 we report the effect of varying N sub-tables for ITR → TaPEx: on the WikiSQL Test set, we get an improvement of 0.4 accuracy points for querying TaPEx on N=10 sub-tables *versus* doing so only on N=1 sub-table. Increasing N up to 20 yields no further improvements. We realise that using a large enough number of sub-tables, one might consider even simpler methods that consider different regions and combinations of the table each time, delegating the selection of the most relevant sub-table to the TableQA system, as per prediction confidence. For this reason, we also compare a naive baseline that uses up to N=10 randomly chopped sub-tables from a given table, without a specific notion of item relevance, combined with TaPEx. For this baseline, we simply sample columns and rows and combine them similar to ITR *mix* until a sub-table exceeds the token budget. Results show that N=10 random sub-tables might allow TaPEx to improve its performance by +2.3% in WikiSQL (*versus* +3.7% improvement from ITR). In WikiTQ, randomly choosing the sub-tables degrades the performance by -3.4% (*versus* +3.7% improvement from ITR). This is because in WikiSQL questions require less interaction between rows/columns and it might

be enough for the system to have visibility of the items containing the gold answer. In WikiTQ, questions require aggregations between different columns and rows, and therefore a random combination of them leads to performance degradation.

We recall that for TaPas, we use N=1 as, due to joint tasks of cell selection and aggregation classification, it is not straightforward to determine the probability of the output making it unfeasible to compare $N > 1$ predictions.

C.4 Computational Cost

Training & Decoding Approach	Training Speed ↑ (iter/sec)	Training Batch Size	Training Time (mins)	Inference Speed ↑ (iter/sec)	Inference Batch Size
TaPEx	3.58	1	460	0.98	16
ITR → TaPEx	3.32	1	480	0.77	4

Table C.5 Training and inference speed for TaPEx and ITR-enhanced TaPEx. We train each model on an A100 machine. Batch size is shown per GPU.

In Table C.5 we report training and inference speed for TaPEx and ITR-enhanced TaPEx. Especially in mix and row-wise ITR variants, the number of sub-tables is generally large (>100), which causes a significant performance bottleneck when dynamically tokenizing the sub-tables within the training/evaluation loop. A potential solution to this problem is that we can calculate the sub-tables at a preprocessing step which does not have any impact on the end-to-end training/inference speed. This is possible as choosing sub-table is not affected by the training updates.

Appendix D

Appendix for Chapter 8

D.1 Table Linearisation

In the retriever component, the input table is linearised into a sequence with separation tokens interleaving the table elements to make the input structure-aware, e.g., "*<SOT> [table title] <EOT> <BOC> mountain peak <SOC> elevation <EOC> <BOR> red slate mountain <SOR> 13,162 ft <EOR> <BOR> ...*".

In the reader component, the TaPEx tokenizer linearises the table with structure-aware separation, for example, "*[HEAD] mountain peak | elevation [ROW] 1 : red slate mountain | 13 , 162 ft [ROW] 2 ...*".

D.2 CLTR and T-RAG Evaluation

In these open-domain TableQA datasets, each question is associated with only one gold table. As a result, Precision@K in retrieval has a certain upper bound at $\frac{1}{K}$. Therefore, evaluating the retriever with Recall@K is more reasonable in this case.

We confirmed with the authors of CLTR and T-RAG that they decomposed tables into single rows and columns to form the table database. In evaluating their systems on the E2E-WTQ dataset, the authors reported some retrieval metrics including Precision@K (P@K) which goes beyond the $\frac{1}{K}$ limit (e.g. T-RAG achieved 0.7806 P@5). This is because they reported a hit for a retrieved row/column as long as it belongs to the gold table. With different setups for table corpus, the retrieval metrics of their systems are not directly comparable. Therefore, we compare Recall@K with BM25 and DPR only, and compare the end-to-end TableQA accuracy with CLTR and T-RAG (which is called Hit@1 in their papers).

Parameter	Value
Negative samples	4 (per positive sample)
Total GPUs	8
Learning rate	0.0001
Optimizer	Adam
Batch size (per device)	8 (DPR) / 6 (LI)
Grad. accum. steps	4
Training steps	6000 (NQ-TABLES) 600 (E2E-WTQ)

Table D.1 Hyperparameters for DPR and LI training.

Parameter	Value (NQ-TABLES)	Value (E2E-WTQ)
Warmup steps	0	
Epochs	20	15
Reader LR	0.00002	0.000015
Retriever LR	0.00001	
LR decay	Linear	None
Optimizer	AdamW	
Total GPUs	8	
Batch size	1 (per device)	
Grad. accum. steps	4	
Weight decay	0.01	
Label smoothing	0.1	

Table D.2 Hyperparameters for LI-RAGE training.

D.3 Technical Details

D.3.1 Hyperparameters

The training hyperparameters are shown in Table D.1, D.2, and D.3. The tuning of hyperparameters was performed on validation performance.

DPR: The dimension of the extracted table/question embeddings is $d = 768$.

LI: The dimension of the extracted table embeddings is $l_t \times d = l_t \times 128$, where l_t depends on the length of input tables. Following Santhanam et al. [264], the dimension of the extracted question embeddings is fixed to $l_q \times d = 32 \times 128$. We pad the questions that have less than l_q tokens.

Parameter	Value
Warmup steps	1000
Epochs	40
Learning Rate	0.00002
LR decay	Linear
Optimizer	AdamW
Total GPUs	8
Batch size	1 (per device)
Grad. accum. steps	4
Weight decay	0.01
Label smoothing	0.1

Table D.3 Hyperparameters for `tapex-large` fine-tuning on WikiTableQuestions for E2E-WTQ.

D.3.2 Indexing and Dynamic Retrieval

DPR. Following Lewis et al. [167], one-dimensional table embeddings are pre-extracted with the DPR model that has been fine-tuned on the retrieval task. The FAISS system [138] is used to index all table embeddings which enables fast nearest neighbour search with sub-linear time complexity. In training LI-RAGE, question embeddings are dynamically extracted from the retriever, and tables with highest scores are retrieved using the precomputed index.
LI. Khattab and Zaharia [146] proposed the first version of ColBERT, and Santhanam et al. [264] introduced ColBERTv2, which is an enhanced version of ColBERT. Santhanam et al. [263] developed an efficient search engine, PLAID, for ColBERTv2, which significantly improved the retrieval latency. We redirect readers to the aforementioned papers for more details. We started from the official ColBERTv2 implementation[1] and refactored the code base. We integrated ColBERTv2 into our training framework, so that fast and dynamic retrieval can be done during end-to-end joint training.

[1] https://github.com/stanford-futuredata/ColBERT