Fine-grained Late-interaction Multi-modal Retrieval for Retrieval Augmented Visual Question Answering (Appendix)

Weizhe Lin, Jinghong Chen,* Jingbiao Mei, Alexandru Coca, Bill Byrne
Department of Engineering
University of Cambridge
Cambridge, United Kingdom CB2 1PZ
{wl356, jc2124, jm2245, ac2123, wjb31}@cam.ac.uk

A Limitations

We chose the Google Search corpus [Luo et al., 2021] for our question-answering system as it provides good coverage of the knowledge needed and is publicly available. However, as noted by the authors of RA-VQA, additional knowledge bases may be required to answer some questions correctly. Future work may address the issue by improving the quality and expanding the coverage of knowledge.

B Ethics Statement

We do not perceive any immediate ethical concerns associated with the misuse of our proposed system. There is a possibility that the trained KB-VQA system 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 the system in live service.

C Data Statistics

Table 1 shows the data statistics of the OK-VQA dataset. Table 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 H, which is used for investigating the retrieval of multi-modal documents.

Table 1: OK-VQA dataset statistics.

Category	Number
train questions	9,009
valid questions	5,046
images	14,055

Table 2: Data statistics of document collections used in retrieval.

Corpus	# of passages
GS for OK-VQA [Luo et al., 2021]	168,306
Wikipedia for OK-VQA	114,637
WIT for OK-VQA (Appendix H)	87,419

D Details of DPR baselines

We build a **DPR** retriever as a baseline for FLMR. We apply the same pre-training strategy, training data, and hyperparameters to construct a multi-modal retriever based on DPR. Particularly, we keep

*Equally contributed as the first author

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

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 $\mathcal{F}_L(\cdot)$ 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:

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

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 [Johnson et al., 2019]) is used to train and retrieve documents with DPR.

E Training and Hyperparameter Details

We use `ColBERTv2` and `openai/clip-vit-base-patch32` to initialize the text-based retriever and vision encoder. For the DPR baseline, we use `facebook/dpr-single-nq-base` to initialize 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 [Kingma and Ba, 2015]. 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 [Hu et al., 2022] 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.

All implementations are released at https://github.com/LinWeizheDragon/Retrieval-Augmented-Visual-Question-Answering.

F 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 [Johnson et al., 2019] (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.

G Computational Cost

We report the computational cost in this section.

Table 3: 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.

	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

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 3, 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. 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 [Santhanam et al., 2022] is required to conduct fast retrieval.

Table 4: 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.

Retriever	Generator	Training Speed (iterations/sec)	Inference Speed (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

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.

H 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 our paper, we present the investigation in this appendix.

Dataset. We select a subset from WIT [Srinivasan et al., 2021], 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 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:

$$\mathbf{D} = [\mathcal{F}_L(d), \mathcal{F}_M(\mathcal{F}_V(I_d))] \in \mathcal{R}^{l_D \times d_L}, \tag{2}$$

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:

$$r(\bar{\mathbf{q}}, \bar{\mathbf{d}}) = r((q, I), (d, I_d)) = \sum_{i=1}^{l_Q} \max_{j=1}^{l_D} \mathbf{Q}_i \mathbf{D}_j^\top \tag{3}$$

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 $\mathcal{F}_M(\mathcal{F}_V(\cdot))$ 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 5, 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 5: FLMR performance when retrieving documents in WIT. Models suffixed by 'uni-modal' only encode document texts, 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**

I Effects of Retrieved Knowledge

It is important to understand the task performance that a base model has attained and the gains from knowledge retrieval. We use the official evaluation metrics from RA-VQA: 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.

$$HSR = \mathbb{1}\{\widehat{y} \in \mathcal{S} \wedge \widehat{y}_{NK} \notin \mathcal{S}\}; \tag{4}$$

Table 6: 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

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 6 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 7: 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 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.

J Case Study

A case study is presented in Fig. 1. It compares the model outputs and provides expalanations to each case.

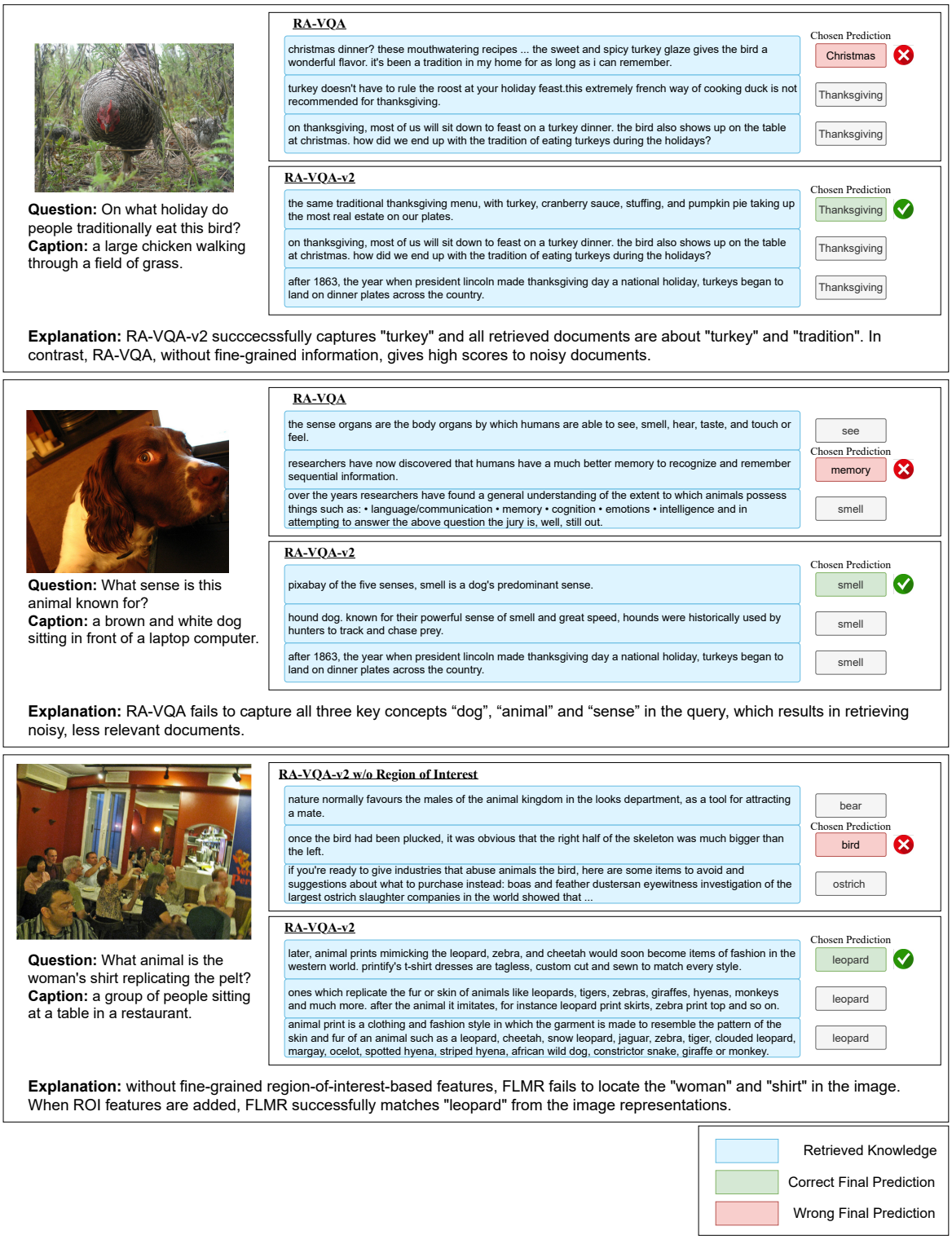

Figure 1: Case study comparing some model variants. Explanations are given to each case. Please zoom in for the best visualization.

References

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. URL https://aclanthology.org/2021.emnlp-main.517.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

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.

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.

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.

Krishna Srinivasan, Karthik Raman, Jiecao Chen, Michael Bendersky, and Marc Najork. Wit: Wikipedia-based image text dataset for multimodal multilingual machine learning. *arXiv preprint arXiv:2103.01913*, 2021.