VDocRAG

CVPR 2025

VDocRAG: Retrieval-Augmented Generation over Visually-Rich Documents

Abstract

- RAG (Retrieval-Augmented Generation)
  - 외부 지식 베이스나 데이터베이스에서 관련 정보를 먼저 검색한 후, 그 정보를 바탕으로 답변을 생성하는 기술
  - 최신 정보 활용, hallucination 감소, 출처 제공, 전문 지식 적용, 정확성 향상
- 기존 방법들은 ..
  - 기존 텍스트 기반 RAG는 OCR이나 파서(예: pdf2text)를 통해 텍스트만 추출해 임베딩하므로, 레이아웃·차트·이미지 같은 비텍스트 정보가 무시되거나 왜곡
  - DocVQA나 ChartQA 같은 작업에서 검색 정확도와 생성 품질을 떨어뜨리며, zero-shot 일반화도 약합니다
  - 개별 데이터셋별로 나뉘어 open-domain 통합 평가 부족
- 이 논문에서는 ..
  - 여러 형식(pdf, ppt 등)과 여러 모달리티(차트, 테이블 등)를 가진 visually-rich 문서에서도 잘하는 RAG 프레임워크를 만들어보자.
  - 기존 텍스트 기반 RAG에서 발생하는 OCR/파서 오류, 레이아웃·도형 정보 유실 등을 피하려고, 문서를 통째로 **이미지 기반 통합 표현**으로 다루는 방식을 제안
  - 다양한 문서 유형과 형식을 포괄하는 오픈 도메인 문서 시각 질의 응답 데이터셋의 첫 통합 컬렉션인 OpenDocVQA 제안

OpenDocVQA Task and Dataset

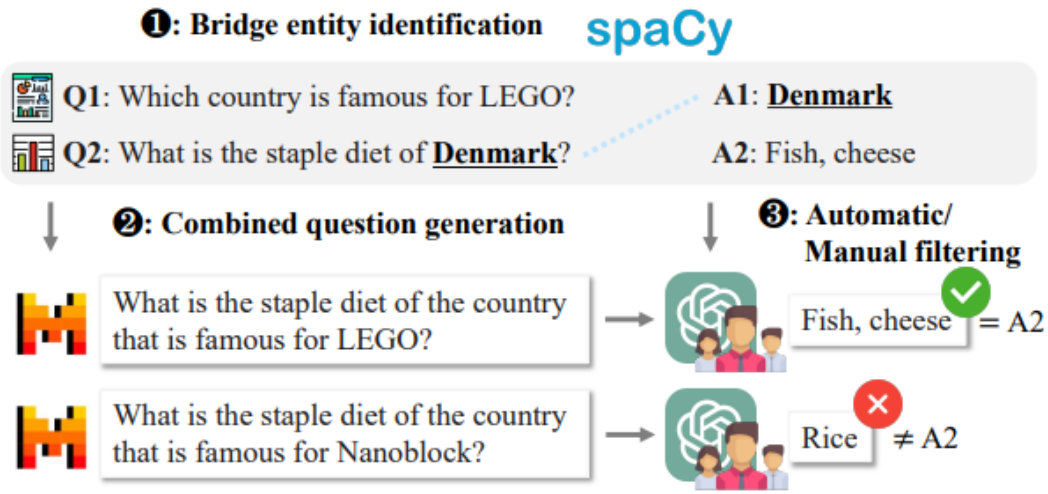
OpenDocVQA BenchMark

- **Visual document retrieval**
  - 대규모 문서 이미지 집합 I와 질문 Q가 주어졌을 때, 관련 top-k 이미지 I′ 를 검색하는 TASK

- **DocumentVQA**
  - 질문 Q와 검색 된 이미지 I'로 답변 A를 생성하는 TASK
- real world 시나리오 반영
  - Single-pool: ChartQA 풀에서만 ChartQA 검색/생성
  - All-pool: 실세계 시나리오처럼 다양한 문서에서 검색/생성

Dataset Collection

- 7개 기존 데이터셋에서 context-independent 질문만 추출
  - Context-dependent ("제목이 뭐야?"): 검색 불가 → retriever 성능 측정 불가능
  - Context-independent ("2024 분기 매출은?"): "2024 매출" 키워드로 정확한 문서 검색 → retriever 정확도 측정 가능
- Open-WikiTable
  - HTML → Wikipedia 스크린샷 이미지로 변환
  - 이미지 기반 task로 재구성.
- MHDocVQA
  - 위에서 수집된 single-hop qa를 기반으로 multi-hop documentVQA 데이터셋 생성



  - 기존
    - text 검색 키워드: "LEGO 주식"

        - LEGO + 음식
        - "LEGO 관련 음식 정보 없음"으로 실패
    - VDocRAG
        - 1st retrieval: "LEGO" → denmark (visual 차트 패턴)
        - 2nd retrieval: "Denmark" → Denmark 음식 표 (visual 표 구조)
    - 다중 문서 간 multi-hop reasoning 테스트

Proposed Model

VDocRAG

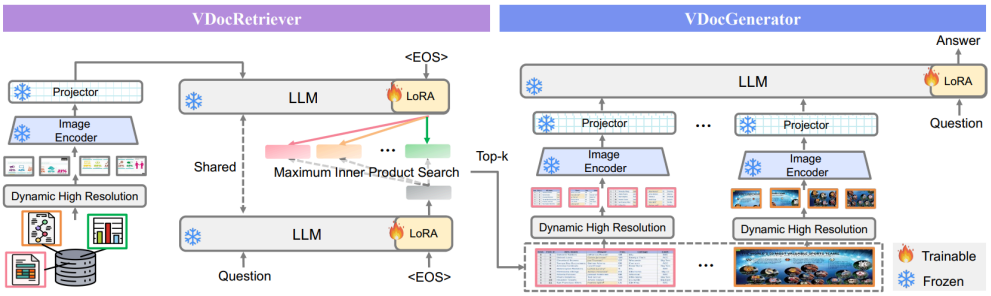


Figure 3. Overview of our VDocRAG model. VDocRetriever retrieves document images related to the question from a corpus of document images, and VDocGenerator uses these retrieved images to generate the answer.

- Dynamic high-resolution image encoding
    - 문서 이미지를 dynamic cropping 적용하여 336x336
    - image → patch → encoder → mlp → z_d
- VDocRetriever
    - large VLM based (phi-3-vision)
    - z_d + <EOS> → LLM → last layer <EOS> = h_d
    - question + <EOS> → LLM → last layer <EOS> = h_q
    - 검색은 cosine-sim 사용 (maximum inner product search)
    - **last hidden state의 <EOS> 에 해당하는 vector를 전체 입력 시퀀스를 대표하는 벡터로 사용하겠다.**

- VDocGenerator
  - 검색된 문서를 encode 하여 question과 concatenate → LLM → Answer

Training

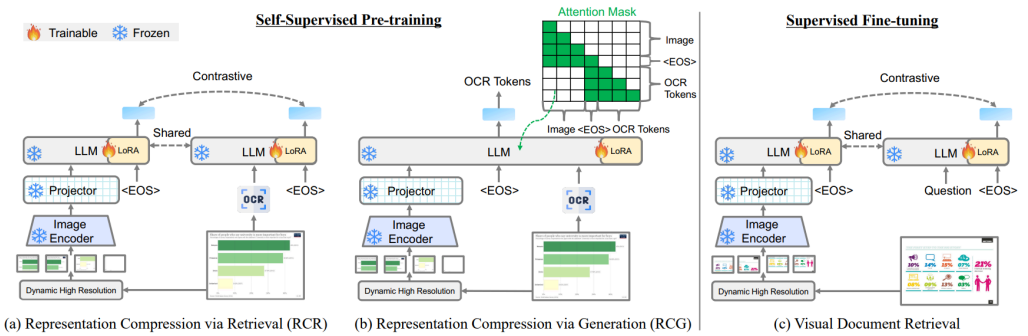


Figure 4. Our pre-training tasks using unlabeled documents and fine-tuning in VDocRetriever. The RCR task retrieves relevant images given corresponding OCR tokens, and the RCG task outputs OCR tokens by paying attention to only the <EOS> token.

LVLM이 제공하는 양질의 이미지 이해도가 <EOS> 벡터에 압축되어 표현될 수 있도록 설계

Self-Supervised Pre-training

- RCR (Representation Compression via Retrieval) - 검색 최적화
  - OCR된 텍스트의 eos hidden vector
  - document image의 eos hidden vector
  - InfoNCE 기반 contrastive loss 사용

$$\mathcal{L}_{\text{RCR}} = -\log \frac{\exp(\text{SIM}(\mathbf{h}_{\text{o}}, \mathbf{h}_{\text{d}^+})/\tau)}{\sum_{i \in \mathcal{B}} \exp(\text{SIM}(\mathbf{h}_{\text{o}}, \mathbf{h}_{\text{d}_i})/\tau)},$$

- RCG (Representation Compression via Generation) - 생성 최적화
  - **이미지 보고 OCR GT를 순차적으로 생성해라**
  - mask 적용하여 OCR 토큰들이 이미지 정보에 직접 의존하지 않고, 압축된 EOS 벡터만 의존하게 만들기
  - OCR 텍스트 토큰 위치에서만 CrossEntropyLoss 적용

- eos, y<i를 보고 yi가 나올 확률

$$\mathcal{L}_{\text{RCG}} = -\frac{1}{L}\sum_{i=1}^{L} \log p(y_i | y_{<i}, \texttt{<EOS>}),$$

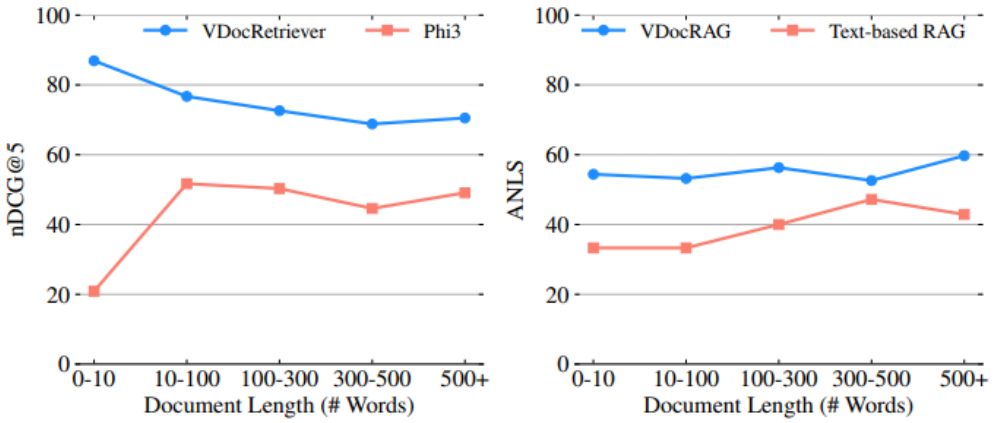
Supervised Fine-tuning

- VDocRetriever
  - OpenDocVQA 데이터셋의 질문-정답문서 쌍을 배치로 묶어서 VDocRetriever를 contrastive learning으로 fine-tuning
- VDocGenerator
  - 학습된 VDocRetriever 사용하여 생성에 사용
  - next-token prediction objective로 학습

Result

Model	Init	Docs	Scale	#PT	#FT	ChartQA		SlideVQA		InfoVQA		DUDE	
						Single	All	Single	All	Single	All	Single	All
Off-the-shelf													
BM25 [52]	–	Text	0	0	0	54.8	15.6	40.7	38.7	50.2	31.3	57.2	47.5
Contriever [22]	BERT [12]	Text	110M	1B	500K	66.9	59.3	50.8	46.5	42.5	21.0	40.6	29.7
E5 [59]	BERT [12]	Text	110M	270M	1M	74.9	66.3	53.6	49.6	49.2	26.9	45.0	38.9
GTE [34]	BERT [12]	Text	110M	788M	3M	72.8	64.7	55.4	49.1	51.3	32.5	42.4	36.0
E5-Mistral [60]	Mistral [23]	Text	7.1B	0	1.85M	72.3	70.0	63.8	57.6	60.3	33.9	52.2	45.2
NV-Embed-v2 [30]	Mistral [23]	Text	7.9B	0	2.46M	75.3	70.7	61.7	58.1	56.5	34.2	43.0	38.6
CLIP [47]	Scratch	Image	428M	400M	0	54.6	38.6	38.1	29.7	45.3	20.6	23.2	17.6
DSE [37]	Phi3V [1]	Image	4.2B	0	5.61M	72.7	68.5	73.0	67.2	67.4	49.6	55.5	47.7
VisRAG-Ret [66]	MiniCPM-V [63]	Image	3.4B	0	240K	**87.2***	75.5*	74.3*	68.4*	71.9*	51.7*	56.4	44.5
Trained on OpenDocVQA													
Phi3 [1]	Phi3V [1]	Text	4B	0	41K	72.5	65.3	53.3	48.4	53.2*	33.0*	40.5*	32.0*
VDocRetriever†	Phi3V [1]	Image	4.2B	0	41K	84.2$_{+11.7}$	74.8$_{+9.5}$	71.0$_{+17.7}$	65.1$_{+16.7}$	66.8*$_{+13.6}$	52.8*$_{+19.8}$	48.4*$_{+7.9}$	41.0*$_{+9.0}$
VDocRetriever	Phi3V [1]	Image	4.2B	500K	41K	86.0$_{+1.8}$	**76.4**$_{+1.6}$	**77.3**$_{+6.3}$	**73.3**$_{+8.2}$	**72.9***$_{+6.1}$	**55.5***$_{+2.7}$	**57.7***$_{+9.3}$	**50.9***$_{+9.9}$

Generator	Retriever	Docs	ChartQA		SlideVQA		InfoVQA		DUDE	
			Single	All	Single	All	Single	All	Single	All
Closed-book										
Phi3	–	–	20.0	20.0	20.3	20.3	34.9*	34.9*	23.1*	23.1*
Text-based RAG										
Phi3	Phi3	Text	28.0	28.0	28.6	28.0	40.5*	39.1*	40.1*	35.7*
Phi3	Gold	Text	36.6	36.6	27.8	27.8	45.6*	45.6*	55.9*	55.9*
VDocRAG (Ours)										
VDocGenerator	VDocRetriever	Image	**52.0**$_{+24.0}$	**48.0**$_{+20.0}$	**44.2**$_{+15.6}$	**42.0**$_{+14.0}$	**56.2***$_{+15.7}$	**49.2***$_{+10.1}$	**48.5***$_{+8.4}$	**44.0***$_{+8.3}$
VDocGenerator	Gold	Image	74.0	74.0	56.4	56.4	64.6*	64.6*	66.4*	66.4*



(a) Retrieval performance (b) QA performance