Unifying Multimodal Retrieval via Document Screenshot Embedding

Anonymous ACL submission

Abstract

In the real world, documents are organized in different formats and varied modalities. Traditional retrieval pipelines require tailored document parsing techniques and content extraction modules to prepare input for indexing. This process is tedious, prone to errors, and has information loss. To this end, we propose *Document Screenshot Embedding* (DSE), a novel retrieval paradigm that regards document screenshots as a unified input format, which does not require any content extraction preprocess and preserves all the information in a document (e.g., text, image and layout). DSE leverages a large vision-language model to *directly* encode document screenshots into dense representations for retrieval. To evaluate our method, we first craft the dataset of Wiki-SS, a 1.3M Wikipedia web page screenshots as the corpus to answer the questions from the Natural Questions dataset. In such a text-intensive document retrieval setting, DSE shows competitive effectiveness compared to other text retrieval methods relying on parsing. For example, DSE outperforms BM25 by 17 points in top-1 retrieval accuracy. Additionally, in a mixed-modality task of slide retrieval, DSE significantly outperforms OCR text retrieval methods by over 15 points in nDCG@10. These experiments show that DSE is an effective document retrieval paradigm for diverse types of documents. Model checkpoints, code, and Wiki-SS collection will be released.

1 Introduction

Information retrieval systems help users access external information from documents in varied modalities, including text, images, charts, and tables. As shown in Figure 1(a), existing document retrieval paradigms typically process these modalities separately. For example, traditional lexical retriever BM25 (Robertson and Zaragoza, 2009) or neural retrievers such as DPR (Karpukhin et al., 2020) rely on extracted text contents from documents. Recent multimodal retrieval (Yang et al., 2023; Wei et al., 2023) leverage both processed text and image units to broaden the scope of retrieval, thus supporting text-image tasks.

However, the existing retrieval paradigms lack a unified encoding process across modalities, leading to two underlying issues. Firstly, preprocessing is not a trivial effort. Specialized processing is required to handle various document types and content modalities, and they are often imperfect. For instance, HTML files in the wild can present significant complexity due to their varied structures, making it difficult for a single tool to parse all information accurately. Similarly, slides and PDFs often require OCR models to extract text and handle other content types like tables and figures separately (Huang et al., 2022; Tanaka et al., 2023). Managing these diverse modalities separately is tedious, and precisely dealing with the long-tailed document appearances in the real world is often impractical. Secondly, this process "breaks" the original appearance of the document, disrupting its visual context and layout integrity. The visual presentation of a document can convey essential information that is difficult to capture through content extraction alone. For example, in addition to the contents of texts and images, the size and position of these elements in a document may encode the importance of the information they contain (Xu et al., 2020; Huang et al., 2022).

To tackle the aforementioned issues, we introduce *Document Screenshot Embedding* (DSE), a new information retrieval paradigm that unifies the varied formats and modalities in a single form for direct document encoding and indexing: screenshot. Unlike the texts and images extracted from a document, screenshots are easy to obtain. More importantly, screenshots naturally preserve all the information in a document. As illustrated in Figure 1(b), DSE directly encodes the screenshot of any given document into a dense representation

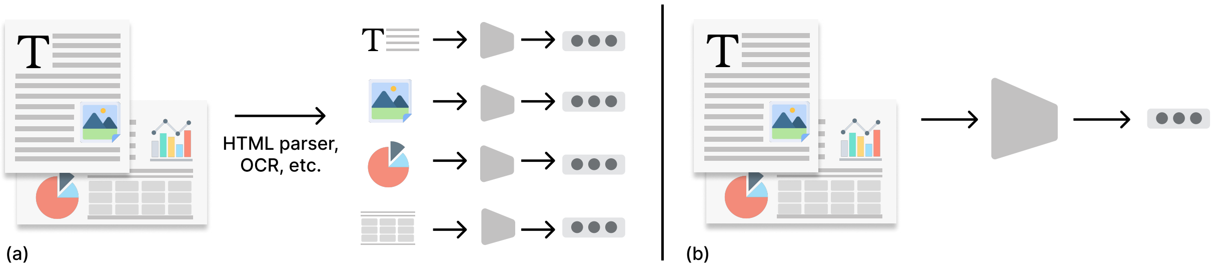

Figure 1: Comparison between (a) existing document retrieval paradigm and (b) our proposed paradigm. DSE bypasses the document parsing and content extraction process, directly encoding the original appearance of documents with multimodal contents into a dense representation for indexing

through a large vision-language model. During search, a user's query is encoded by a language model to locate the nearest document embeddings.

We conduct empirical studies to demonstrate that DSE is effective for document retrieval. Specifically, we conduct experiments on two types of document retrieval settings: text-intensive setting and text-image mixed one. For the former, we collect 1.3 million Wikipedia web page screenshots as our corpus and fine-tune a large vision-language model as a bi-encoder to conduct dense retrieval on questions in the NQ dataset (Kwiatkowski et al., 2019). Experimental results show that DSE outperforms the traditional text-based retrieval method BM25 by 17 points in top-1 retrieval accuracy on NQ questions and is competitive with text-based dense retrieval methods in a text-oriented evaluation. This experiment indicates that DSE can sufficiently encode the textual information in a screenshot. For the image-text mixed setting, we use slide retrieval. We turn the existing SlideVQA (Tanaka et al., 2023) dataset into an open-domain retrieval setting, where models are required to retrieve relevant slides from a pool of 50k slides for given questions. Results show that DSE outperforms all text-based retrieval methods which rely on OCR (including BM25 and dense text retrieval) by over 15 points in nDCG@10.

2 Related Work

2.1 Neural Document Retrieval

Traditional document retrieval methods such as TF-IDF and BM25 (Robertson and Zaragoza, 2009) represent text as bag-of-words representations and conduct efficient search over an inverted index. Recent neural retrieval methods represented by DPR (Karpukhin et al., 2020), proposed to use to finetune pretrained neural networks such as BERT (Devlin et al., 2019) to encode query and document separately into dense semantic vectors in a bi-encoder architecture. The effectiveness of text dense retriever has been boosted in recent years by various training strategies such as data augmentation (Xiong et al., 2021; Lin et al., 2023; Xiao et al., 2023), pretraining (Izacard et al., 2021; Gao and Callan, 2022; Wang et al., 2023a), and distillation (Lin et al., 2021; Ren et al., 2021). With the growth of large language models finetuning LLM as text embedding demonstrated further improvement in both in-domain and out-domain retrieval effectiveness (Ma et al., 2023; Wang et al., 2023b; Lee et al., 2024).

Besides text retrieval, prior multi-modal retrieval studies (Wei et al., 2023; Koukounas et al., 2024) have explored retrieval across various combinations of text and image inputs for queries and documents. These approaches aim to bridge the gap between different modalities, enabling more comprehensive retrieval systems.

Existing text and multi-modal retrieval works assume that the datasets are well pre-processed, where text and image data are carefully extracted and organized for model inputs. However, this is not always true in real-world scenarios where documents are often unstructured and diverse. In this work, we consider the document retrieval tasks that begin with the original look of documents.

2.2 Large Vision-Language Model

Large language models (LLMs) like ChatGPT (OpenAI, 2022) and LLaMA (Touvron et al., 2023), pre-trained on massive corpora and fine-tuned to follow user instructions, have shown success in various natural language generation tasks (Wei et al., 2022). Recent advancements have integrated vision capabilities into LLMs, enabling them to process both text and images simultaneously. Commercial models like GPT-4V (OpenAI, 2024) and open-source models such as LLaVA (Liu et al., 2023) exhibit strong performance. Building upon LLaVA, recent works such as LLaVA-NEXT (Liu et al., 2024a), Idefics2 (Laurençon et al., 2024), and Phi-3-vision (Abdin et al., 2024) have further improved performance. They enable the processing of higher-resolution images and handle more challenging vision-language tasks, such as OCR (Liu et al., 2024a,b). Inspired by the capabilities of large vision-language models, our work pioneers its application in document retrieval tasks.

2.3 Document Retrieval Datasets

Commonly used text retrieval datasets such as MS MARCO (Bajaj et al., 2018), Wikipedia-NQ (Karpukhin et al., 2020), and BEIR (Thakur et al., 2021) are typically released in well-preprocessed and cleaned text contents. Similarly, multi-modal retrieval datasets like AToMIC (Yang et al., 2023) and m-BEIR (Wei et al., 2023) have text and images extracted from their sources and separately stored.

On the other hand, existing datasets designed for question-answering tasks based on document images include DocVQA (Mathew et al., 2021), VisualMRC (Tanaka et al., 2021), WebSRC (Chen et al., 2021), and InfographicVQA (Mathew et al., 2022). These datasets contain document images paired with questions, focusing on reading comprehension evaluation where a ground truth document image is provided for each question. Besides, the document image pools in these datasets are relatively small, typically comprising only a few thousand images.

To evaluate multi-modal document retrieval in a large scale, we crafted a text-intensive image corpus called Wiki-SS, containing 1.3 million Wikipedia page screenshots which support retrieval evaluation in large scale. Additionally, we convert SlideVQA (Tanaka et al., 2023) dataset, a visual QA dataset, into an open-domain slide retrieval dataset, consisting of a corpus of 50K slides.

3 Method

3.1 Task Definition

Given a query Q and a corpus $\mathcal{C}$ consisting of documents $\{D_1, D_2, ..., D_n\}$, the task of document retrieval is to identify the k documents that are most relevant to the query Q, with $k \ll n$. This relevance is determined using a similarity metric $\text{Sim}(Q, D) \in \mathbb{R}$. Note that in this work, the screenshot "document" is a complete information snippet (e.g. a web article, a PDF page). This is different from some of the previous retrieval work, where the term "document" denotes arbitrary information snippets like sentences or passages. For queries, we only consider the text inputs similar to the traditional search setting. We leave the exploration of handling image queries for future work.

3.2 Document Screenshot Embedding

We adopt a bi-encoder architecture for dense retrieval, where a document screenshot and user text query are encoded into dense vectors using a vision and text encoder, respectively. We can naively apply the vision and text encoders from CLIP (Radford et al., 2021) to our task; however, in our experiment, we observe that the vision encoder cannot encode screenshots with more fine-grained information; thus, we propose to use large vision language models as the document screenshot encoder.

Visual Encoder When a document screenshot D is provided, it is first processed by a vision encoder E_v to generate a sequence of latent representations. The length of the sequence is determined by the image tokenizer of the vision encoder. We take `clip-vit-large-patch14-336`[1] as an example. Any given screenshot is first converted to an image with 336×336 pixels and then divided into 24×24 patches (i.e., 576 patches in total), each of which consists of 14×14 pixels. Each patch is flattened and mapped to a patch embedding with a trainable linear projection. The patch embeddings are encoded into latent representations with a vision encoder. However, if a screenshot contains many texts (e.g., Wikipedia webpage), the 576 patch latent embeddings may not capture the fine-grained textual information in the screenshot.

Vision Language Model In order to address the aforementioned issue, we leverage a large vision language model, Phi-3-vision[2], which uses the same image tokenizer from `clip-vit-large-patch14-336` but can represent an image with more patches by cropping it into sub-images. For example, given a screenshot, we can choose to divide it into $(C_x \times 24) \times (C_y \times 24)$ patches. The given screenshot is converted to an image with $(C_x \times 336) \times (C_y \times 336)$ pixels and cropped into $C_x \times C_y$ sub-images, each of which has 336×336 pixels. Similarly, each sub-image is

[1] ViT-Large

[2] Phi-3-vision

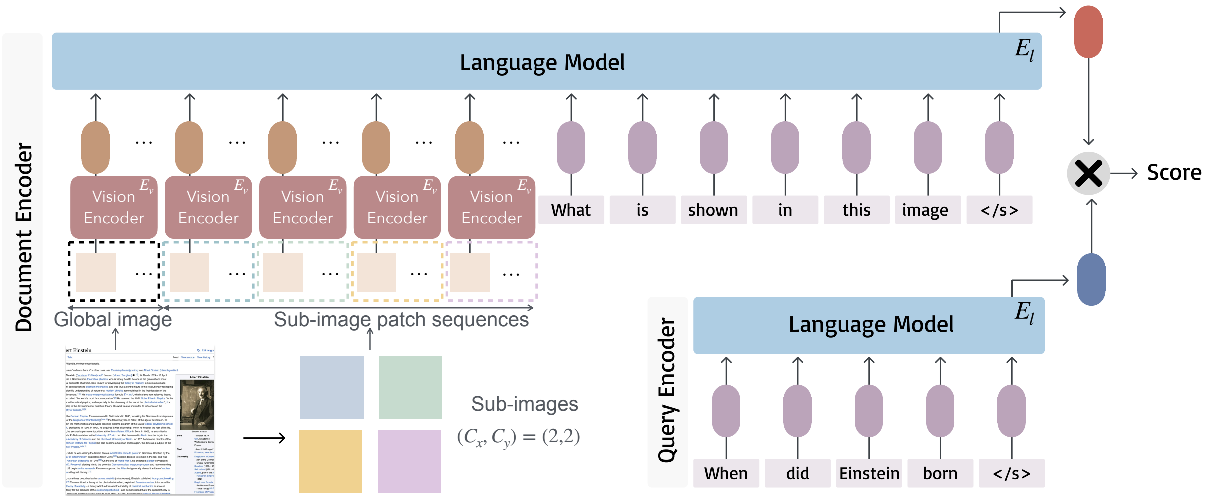

Figure 2: Overview of DSE encoder architecture. DSE adopts a bi-encoder architecture, where the document tower encodes the document screenshot into dense vector by taking vision input and the query tower encodes the query by taking text input. Document and query encoders share the same language model.

encoded into 576 patch latent representations independently. Note that Phi-3-vision further converts the whole screenshot into 336 × 336 pixels and encodes them into an additional 576 patch latent representations to capture the global information, resulting in $(C_x \times C_y + 1) \times 576$ patch latent representations in total, as depicted in left side of Figure 2. Also, every four patch latent representations are merged into one for language model inputs. This process yields $(C_x \times C_y + 1) \times \frac{576}{4}$ patch latent embeddings as the input for the language model E_l. In Section 5.3, we will show that encoding a screenshot into more patch latent embeddings (increasing C_x and C_y) helps capture more fine-grained information in the screenshot but sacrifices screenshot document encoding efficiency.

The encoded patch latent embeddings are concatenated with a text prompt as the input to the subsequent language model: *"<s><img> What is shown in this image?</s>"*. Here, the <img> token is a special placeholder token and is replaced by the sequence of patch latent embeddings. In order to better aggregate information using a language model with uni-directional attention, following Ma et al. (2023), we use the end-of-sequence token </s> embedding from the last hidden state as the document screenshot embedding:

$$V_d = E_l(E_v(D), \text{prompt})[-1]$$

Contrastive Learning The similarity between the query and the document is computed as the cosine similarity between their embeddings:

$$\text{Sim}(Q, D) = \frac{V_q^\top V_d}{\|V_q\| \cdot \|V_d\|}.$$

During training, our embedding model is optimized using the InfoNCE loss:

$$\mathcal{L}(Q, D^+, D_{\text{N}}) = -\log p(D = D^+ \mid Q)$$
$$= -\log \frac{\exp(\text{Sim}(Q, D^+)/\tau)}{\sum\limits_{D_i \in \{D^+\} \cup D_N} \exp(\text{Sim}(Q, D_i)/\tau)},$$

where D^+ denotes the positive document. D_{N} represents a set of negative documents that are irrelevant to the query Q, including hard negatives and in-batch negatives. τ is a temperature parameter set to 0.02 in our experiments. Note that we only consider text queries, which are directly input to the language model using template *f"<s>{query}</s>"* and the last hidden state of *</s>* is used as the query embedding, $V_q = E_l(Q)[-1]$.

4 Experiment Setup

4.1 Web-Page Retrieval

Dataset We construct the Wiki-SS dataset, using the Selenium Python toolkit[3] to access English Wikipedia pages through URLs and automatically take screenshots. The screenshots are taken with a window size of 980 x 980 pixels to ensure adequate coverage of the core content. The screenshot creation process is conducted over a span of four

[3] https://pypi.org/project/selenium/

days, from May 20 to May 23, 2024. Note that storing the entire collection of Wikipedia screenshots would require over 2TB of storage in PNG format. In order to make Wiki-SS more manageable for research purposes, we downsize the corpus by filtering out the web pages which are considered "easy negative samples" for all the questions in the train, dev and test sets from Natural Questions (Kwiatkowski et al., 2019). Specifically, we perform BM25 search for each question to retrieve the top 50 documents over the text corpus. The retrieved documents are pooled together as our final corpus. Note that we concatenate each question and its corresponding ground truth answers as a query for BM25 search to ensure that positive and hard negative documents for each question are included in the downsized corpus. As a result, we obtain a collection of 1,267,874 Wikipedia screenshots for our experiments.

To compare with text-based retrieval baselines, we create a text version Wikipedia collection which mirrors the collection of Wiki-SS. Given the significant updates and changes to Wikipedia pages over time, the existing Wikipedia dumps (Karpukhin et al., 2020; Izacard et al., 2024) cannot be used as a fair comparison. Thus, we re-process the Wikipedia text contents based on the May 20, 2024 dump[4] uses Wikipeida parsing tool `mwparserfromhell`. For each document in the text corpus, we use the first 500 words of each document, mirroring the corpus in Wiki-SS, where each screenshot covers only the first-page content.

Training Data We create the training data by taking the questions in the NQ train split as queries and using BM25 to retrieve the top-50 relevant documents over the text corpus for each question. A document candidate (either in screenshot or text) is considered positive when the corresponding text contains the answers for the question. Otherwise, the document is considered a hard negative candidate. We drop the training example if either the positive or negative candidate list is empty, resulting in 49,095 training examples of triplets of query, positive documents and hard negative documents.

Evaluation We evaluate the in-domain effectiveness of retrievers using the 3,610 NQ test set questions. Consistent with previous practices in evaluating retrieval effectiveness on QA datasets (Karpukhin et al., 2020), we use top-k retrieval accuracy as the metric. A question is considered correctly answered if one of the candidate documents contains an exact match of the answer string in the corresponding text content. We follow the same method for computing exact match accuracy as Karpukhin et al. (2020).

4.2 Slide Retrieval

Dataset The original SlideVQA (Tanaka et al., 2023) data is designed for document visual question answering. It contains 14.5k QA pairs and 52k slide images in total. The images contain various text formats, layouts, and visual content such as plots and charts. Given a question, the original task is to select the most relevant slides among the same deck with up to 20 slides and then answer the question based on the selected slides. The document selection process is in the form of reranking and classification. In order to support the evaluation of document retrieval, we modify the SlideVQA to an open-domain retrieval task, where the task is to retrieve k most relevant slide from the entire pool of slide images. After our processing (e.g. removing the slides that fail to download, and questions that do not have evidence slides available), SlideVQA-open contains 50,714 slide images (screenshots) in its corpus. We also create a corresponding text-based corpus for comparison with text retrievers using `pytesseract` OCR toolkit to extract text from slides.

Training Data We create the training data based on the original train split of SlideVQA, the annotated evidence slides for a given question are considered positive documents, and the other slides within the same deck are considered as hard negative documents. This process leads to 10,290 training examples in total.

Evaluation We construct the SlideVQA-open evaluation set using the 2,136 questions in the test set of SlideVQA. We evaluate the models' retrieval effectiveness using nDCG@10 and Recall@10. In the following sections, mentions of SlideVQA refer to the open-domain retrieval setup.

4.3 Implementation Details

We implement DSE by modifying the Tevatron toolkit (Gao et al., 2023), with the model initialized using Phi-3-vision (Abdin et al., 2024), one of the state-of-the-art open-source large vision-language models with 4 billion parameters. This model is

[4] `https://huggingface.co/datasets/legacy-datasets/wikipedia`

Retriever	**Document**	**NQ**				**SlideVQA-open**	
		Top 1	Top 5	Top 10	Top 20	nDCG@10	Recall@10
BM25	Text	29.5	52.6	61.3	67.3	55.8	63.7
DPR		42.3	63.9	69.7	74.3	47.4	57.9
E5		47.6	68.6	73.8	77.6	59.3	69.6
Phi-3		50.6	70.9	75.8	79.5	59.0	69.5
CLIP	Screenshot	35.1	50.8	57.7	64.8	61.7	74.7
DSE		46.2	68.5	73.7	77.6	75.3	84.6

Table 1: Supervised retrieval effectiveness comparison. DSE and CLIP directly encode document screenshots while the other text-based retrieval models encode the extracted text from documents.

recognized for its effective and efficient trade-off in performance. To train the model, we employ memory-efficient techniques such as LoRA (Hu et al., 2022), FlashAttention (Dao, 2024), and DeepSpeed (Rasley et al., 2020). The model is trained with a batch size of 128 for one epoch on Wikipedia webpage retrieval and trained with a batch size of 64 for two epochs for slide retrieval. The model weights are shared between the language models for document screenshot and query encoding. In both tasks, each training query is paired with one positive document and one hard negative document. We set $(C_x, C_y) = (4, 4)$ by default; that is, the document screenshots are resized to 1344x1344 pixels and cropped into 4x4 sub-images. The training process is conducted on two A100 80GB GPUs. During inference, the embeddings are indexed using a Flat Faiss index (Douze et al., 2024) for exact nearest neighbor search.

4.4 Baselines

We compare DSE against the following document retrieval methods based on text input: (1) BM25: a traditional text retriever based on lexical representation. (2) DPR: we follow the same setting as the DPR work (Karpukhin et al., 2020), initialize dense retriever with BERT-base, and finetuned on our training data based on text input. (3) E5: Similar to DPR, we finetune the unsupervised E5-base model (Wang et al., 2022), which has BERT further pretrained with constrastive learning based on web data. (4) Phi-3: we use the same model initialization and configuration as DSE but only fine-tune the component of the language model as a text-based dense retriever. Additionally, we compare the fine-tuned CLIP model, whose image encoder is also initialized by ViT-large (the same as DSE) but only supports a fixed length of patch sequence; i.e., $(C_x, C_y) = (1, 1)$. See Appendix A.2 for detailed hyper-parameters of DSE and baselines.

5 Experimental Results

5.1 Supervised Retrieval Effectiveness

Table 1 presents the models' retrieval effectiveness in the supervised setting, where models are fine-tuned on NQ or SlideVQA training queries and evaluated on the corresponding evaluation set. For the Wikipedia webpage retrieval task, DSE demonstrates significant improvements over the traditional text-based retrieval method BM25. Specifically, DSE achieves 46.2% and 77.6% in top-1 and top-20 retrieval accuracy, which are 17 points and 10 points higher than BM25, respectively. This indicates that DSE can effectively encode text-intensive documents in the format of screenshots for retrieval. When compared with neural text retrieval methods, DSE outperforms smaller model DPR and performs on par with E5. Phi-3, which uses the same language model as DSE (with 4 billion parameters), achieves approximately 4 points higher top-1 retrieval accuracy than DSE. This suggests that existing vision language models still cannot fully capture the text content in a screenshot.

In the slide retrieval task, where the documents include a mix of text and visual content, we observe DSE significantly outperforms (i.e., over 15 points in both nDCG@10 and Recall@10) all the text retrieval baselines that rely on OCR content extraction. This highlights the risk of information loss in the content extraction step, where OCR is only able to extract text content, thereby losing the visual elements of the documents. Notably, DPR, a neural retrieval method, fails to outperform BM25 in this task. This may be due to the varied layouts of slides, which pose additional challenges for text content extraction and result in noisy text input

Zero-Shot	**TriviaQA**		**SlideVQA-open**	
Retriever	Top 1	Top 10	nDCG@10	Recall@10
BM25	47.4	71.0	55.8	63.7
DPR	37.3	65.5	29.5	39.7
E5	46.9	73.1	42.6	54.4
Phi-3	57.1	78.1	49.7	62.1
CLIP	37.3	65.6	48.4	61.6
DSE	50.3	75.2	64.0	76.1

Table 2: Zero-shot retrieval effectiveness comparison. Models are trained on Wiki-SS with NQ questions and evaluated on TriviaQA questions and slide retrieval task.

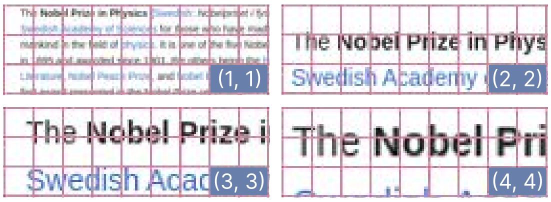

Figure 3: A snapshot of a Wikipedia webpage divided by different numbers of patches (red small squares). As the number of patches increases, each patch can capture more fine-grained text information in the screenshot. (C_x, C_y) means the image are divided into $C_x \times C_y$ sub-images; then converted into $(C_x \times 24) \times (C_y \times 24)$ patches. See more detail in Section 3.2 and Figure 2.

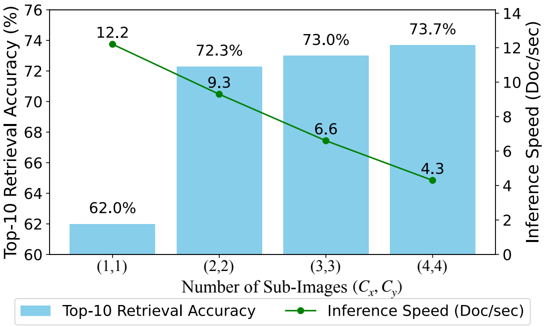

Figure 4: Trade-off between effectiveness and efficiency of DSE with varying numbers of crops for input images. The inference speed is measured on a single H100 GPU with BF16 precision and FlashAttention enabled.

for text neural retrieval fine-tuning. By contrast, DSE bypasses the stage of text content extraction and directly encoding document screenshots, which preserves more information for retrieval.

Finally, DSE outperforms CLIP even though they use the same backbone of the vision transformer to digest the document screenshots. For NQ, DSE surpasses CLIP by 11.1 points in top-1 accuracy, and for SlideVQA, DSE achieves 12.6 points higher in nDCG@10. We contribute the effectiveness gain to the large vision-language model encoder, which as we will show in Section 5.3, has the capacity to handle more fine-grained information in a screenshot and possibly enhanced semantic understanding.

5.2 Zero-Shot Retrieval Effectiveness

In this section, we further evaluate the generalization capability of DSE. Specifically, we apply the models fine-tuned on NQ questions to retrieve answers for TriviaQA questions (Joshi et al., 2017) over the Wiki-SS (or the corresponding Wiki text) corpus, assessing their ability to generalize across different query distributions. Additionally, we evaluate the NQ fine-tuned models on the SlideVQA dataset to examine cross-task generalization.

As shown in Table 2, on TriviaQA, the text retriever based on LLM (i.e., Phi-3) achieves the best zero-shot effectiveness with a top-1 retrieval accuracy of 57.1%. Both DPR and CLIP show lower zero-shot effectiveness, being outperformed by BM25 by approximately 10 points. In contrast, DSE achieves a top-1 retrieval accuracy of 50.3%, which is 3 points higher than BM25. This indicates that DSE has relatively good zero-shot effectiveness across different query distributions but with room for improvement.

On the slide retrieval task, we observe that DSE shows the best effectiveness among all. Specifically, DSE outperforms BM25 by 8 points in terms of nDCG@10, while all the other text-based methods underperform BM25. This result shows that even though DSE is only fine-tuned on the Wikipedia webpage retrieval task, where text is the main content, it is still able to encode document information beyond text. This demonstrates the potential of DSE in handling diverse document types and tasks without needing task-specific training.

5.3 Impacts of Patch Sequence Length

As we discussed in Section 3.2, each screenshot input to DSE is cropped into $C_x \times C_y$ sub-images and encoded as a sequence of patches. Thus, increasing the number of crops yields a more lengthy patch input sequence, which incurs more computation cost for document encoding. On the other hand, increasing the number of crops results in patches with more fine-grained visual information, as illustrated in Figure 3. In the setting of $(C_x, C_y) = (1, 1)$, each patch contains multiple words, while in the setting of $(C_x, C_y) = (4, 4)$, a single letter is covered by two patches. This leads to a trade-off between the efficiency and quality of document en-

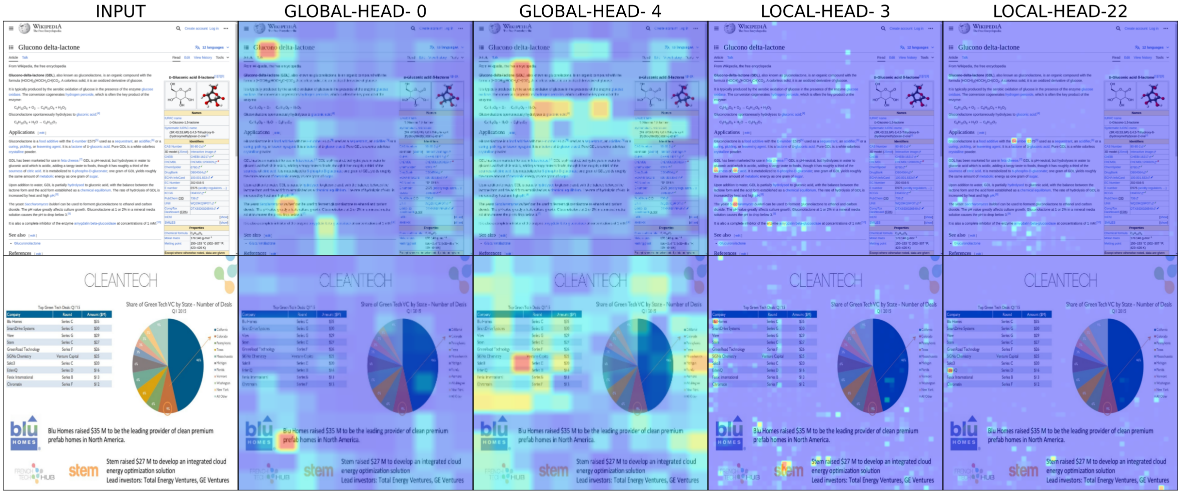

Figure 5: Case study on two examples in Wikipedia and SlideQA. We visualize the multi-head attention from the fine-tuned embedding to the image patches at the last layer. GLOBAL-HEAD is the attention head to the coarse image features (336×336), while the LOCAL-HEAD is the attention head to more fine-grained image features after cropping (16×336×336). We verify that the textual information is indeed extracted from the screenshots.

coding. We study this trade-off by training DSE with different numbers of crops and evaluate the corresponding retrieval effectiveness and document encoding speed (Doc/sec) on the Wiki-SS task for NQ questions.

We plot the efficiency and effectiveness in Figure 4. When cropping the image into 4x4 sub-images for more fine-grained patch encoding, the top-10 retrieval accuracy increases from 62.0% to 73.7%, indicating that finer granularity helps the model better understand and encode the document screenshot. However, this comes at the cost of computational efficiency. As the number of sub-images increases, the sequence length of the model's input grows, resulting in longer encoding times. The document encoding speed decreased from 12.2 documents per second with 1×1 sub-images to 4.3 documents per second with 4×4 sub-images as input. Finally, the experiment suggests that using $(C_x, C_y) = (2, 2)$ or $(3, 3)$ offers a good balance between retrieval effectiveness and computational efficiency.

5.4 Case Study

We conducted a case study to verify whether the fine-tuned embeddings effectively utilize the core semantic information in the screenshots. Figure 5 presents the attention visualization of two examples from Wiki-SS and SlideVQA. We used the Phi-3-vision model fine-tuned on NQ as the backbone and extracted the multi-head attention of the last token embedding to the image patches at the final layer. The image patches contain both global and local features: Global features are tokenized from the resized full image input (336×336), while local features are derived from crops when the image is resized to 1344×1344 and then cropped into 4×4 sub-images before encoding. For both examples, the global attention heads appear to focus on general information, such as images, logos, titles, and sections. In contrast, the local attention heads concentrate on finer details in the screenshots, such as individual letters and keywords, which are crucial for retrieval. This qualitative evidence suggests that DSE can effectively capture information from various modalities within the screenshots, thereby enhancing its retrieval capabilities.

6 Conclusion

In this paper, we introduced DSE, a novel information retrieval paradigm that leverages screenshots to simplify the document retrieval process. By bypassing traditional preprocessing steps and directly encoding documents with a vision-language model, DSE offers a unified approach to handling varied document modalities. We empirically show that DSE outperforms traditional retriever and OCR-based methods on varied document retrieval tasks, such as webpage and slide retrieval. This underscores the potential of DSE to enhance document retrieval in diverse real-world applications. Future developments could refine encoding techniques and adapt to different document types, setting new standards for multi-modal information retrieval.

7 Limitations

This work has several limitations that warrant further exploration. Firstly, while we evaluated DSE on Wikipedia webpage retrieval and slide retrieval datasets, there remains a gap in its effectiveness for more general-purpose document retrieval tasks, such as those involving PDFs or web pages with highly varied structures and content. Future work can consider multi-task training across diverse document types and content. Additionally, combining our method with extracted text and image contents could make DSE more versatile for general retrieval tasks. Secondly, our current approach relies solely on supervised fine-tuning. However, research in text retrieval has shown that contrastive pretraining can significantly improve retriever effectiveness. Investigating whether such pretraining methods can enhance DSE's performance is a promising direction for future research. Thirdly, the reliance on visual data introduces challenges in environments where such data is of low quality. Blurry or low-resolution screenshots may degrade the effectiveness of DSE. Conversely, processing very high-resolution images can reduce computational efficiency. We leave further explore the balance of image quality and computational efficiency as future work.

References

Marah Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Qin Cai, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Yen-Chun Chen, Yi-Ling Chen, Parul Chopra, Xiyang Dai, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Victor Fragoso, Dan Iter, Mei Gao, Min Gao, Jianfeng Gao, Amit Garg, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Ce Liu, Mengchen Liu, Weishung Liu, Eric Lin, Zeqi Lin, Chong Luo, Piyush Madan, Matt Mazzola, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Xin Wang, Lijuan Wang, Chunyu Wang, Yu Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Haiping Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Jianwei Yang, Ziyi Yang, Yifan Yang, Donghan Yu, Lu Yuan, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv:2404.14219*.

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. 2018. Ms marco: A human generated machine reading comprehension dataset. *arXiv:1611.09268*.

Xingyu Chen, Zihan Zhao, Lu Chen, JiaBao Ji, Danyang Zhang, Ao Luo, Yuxuan Xiong, and Kai Yu. 2021. WebSRC: A dataset for web-based structural reading comprehension. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4173–4185, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Tri Dao. 2024. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library. *arXiv:2401.08281*.

Luyu Gao and Jamie Callan. 2022. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2843–2853, Dublin, Ireland. Association for Computational Linguistics.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Tevatron: An efficient and flexible toolkit for neural retrieval. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 3120–3124, New York, NY, USA. Association for Computing Machinery.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and

Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 4083–4091, New York, NY, USA. Association for Computing Machinery.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2021. Unsupervised dense information retrieval with contrastive learning.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2024. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. 2017. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1601–1611, Vancouver, Canada. Association for Computational Linguistics.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781, Online. Association for Computational Linguistics.

Andreas Koukounas, Georgios Mastrapas, Michael Günther, Bo Wang, Scott Martens, Isabelle Mohr, Saba Sturua, Mohammad Kalim Akram, Joan Fontanals Martínez, Saahil Ognawala, Susana Guzman, Maximilian Werk, Nan Wang, and Han Xiao. 2024. Jina clip: Your clip model is also your text retriever. *arXiv:2405.20204*.

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. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Hugo Laurençon, Léo Tronchon, Matthieu Cord, and Victor Sanh. 2024. What matters when building vision-language models? *arXiv:2405.02246*.

Chankyu Lee, Rajarshi Roy, Mengyao Xu, Jonathan Raiman, Mohammad Shoeybi, Bryan Catanzaro, and Wei Ping. 2024. Nv-embed: Improved techniques for training llms as generalist embedding models. *arXiv:2405.17428*.

Sheng-Chieh Lin, Akari Asai, Minghan Li, Barlas Oguz, Jimmy Lin, Yashar Mehdad, Wen-tau Yih, and Xilun Chen. 2023. How to train your dragon: Diverse augmentation towards generalizable dense retrieval. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 6385–6400, Singapore. Association for Computational Linguistics.

Sheng-Chieh Lin, Jheng-Hong Yang, and Jimmy Lin. 2021. In-batch negatives for knowledge distillation with tightly-coupled teachers for dense retrieval. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 163–173, Online. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024a. Llava-next: Improved reasoning, ocr, and world knowledge. https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yuliang Liu, Zhang Li, Biao Yang, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. 2024b. On the hidden mystery of ocr in large multimodal models. *arXiv:2305.07895*.

Xueguang Ma, Liang Wang, Nan Yang, Furu Wei, and Jimmy Lin. 2023. Fine-tuning llama for multi-stage text retrieval. *arXiv:2310.08319*.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C.V. Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 1697–1706.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

OpenAI. 2022. Introducing chatgpt. https://openai.com/blog/chatgpt.

OpenAI. 2024. Gpt-4 technical report. *arXiv:2303.08774*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *arXiv:2103.00020*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 3505–3506, New York, NY, USA. Association for Computing Machinery.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, QiaoQiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2021. RocketQAv2: A joint training method for dense passage retrieval and passage re-ranking. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2825–2835, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3:333–389.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. 2023. Slidevqa: A dataset for document visual question answering on multiple images. In *AAAI*.

Ryota Tanaka, Kyosuke Nishida, and Sen Yoshida. 2021. Visualmrc: Machine reading comprehension on document images. In *AAAI*.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. 2021. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *arXiv:2302.13971*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training. *arXiv preprint arXiv:2212.03533*.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2023a. SimLM: Pre-training with representation bottleneck for dense passage retrieval. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2244–2258, Toronto, Canada. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023b. Improving text embeddings with large language models. *arXiv:2401.00368*.

Cong Wei, Yang Chen, Haonan Chen, Hexiang Hu, Ge Zhang, Jie Fu, Alan Ritter, and Wenhu Chen. 2023. Uniir: Training and benchmarking universal multimodal information retrievers. *arXiv:2311.17136*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *arXiv: 2309.07597*.

Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul N. Bennett, Junaid Ahmed, and Arnold Overwijk. 2021. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. Layoutlm: Pre-training of text and layout for document image understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1192–1200, New York, NY, USA. Association for Computing Machinery.

Jheng-Hong Yang, Carlos Lassance, Rafael Sampaio de Rezende, Krishna Srinivasan, Miriam Redi, Stéphane Clinchant, and Jimmy Lin. 2023. Atomic: An image/text retrieval test collection to support multimedia content creation. *arXiv:2304.01961*.

A Appendix

A.1 Dataset Licences

- **NQ**: Apache License 2.0
- **TriviaQA**: Apache License 2.0
- **SlideVQA**: SOFTWARE LICENSE AGREEMENT FOR EVALUATION
- **Wikipedia**: Creative Commons Attribution Share Alike, GNU Free Documentation License family.
- **Wiki-SS**: Creative Commons Attribution Share Alike, GNU Free Documentation License family.

A.2 Hyper-Parameters for Training

Please see Table 3 for details.

A.3 AI assistants usage

GPT4o is used during the writing to capture grammar errors and format tables.

Method	**DPR**	**E5**	**Phi3**	**CLIP**	**DSE**
Model Init	google-bert/bert-base-uncased	intfloat/e5-base-unsupervised	microsoft/Phi-3-vision-128k-instruct	openai/clip-vit-large-patch14-336	microsoft/Phi-3-vision-128k-instruct
License	Apache 2.0	MIT License	MIT License	MIT License	MIT License
# of Parameters	110 M	110 M	4B	430 M	4B
Backbone Modality	text	text	text or vision	text XOR vision	text OR vision
Learning Rate	1e-5	1e-5	1e-4	1e-5	1e-4
GPU	2xA100 80G	2xA100 80G	2xA100 80G	2xA100 80G	2xA100 80G
Per Device Batch Size	64	64	8	16	8
Hard Neg Per Query	1	1	1	1	1
Gradient Accumulation	1	1	8 (4)	4	8 (4)
Total Batch Size	128	128	128 (64)	128	128 (64)
Pooling	cls	mean	eos	mean	eos
Temperature	1	0.02	0.02	0.02	0.02
Normalize	False	True	True	True	True
Epochs	40	40	1 (2)	10	1 (2)
LoRA	False	False	True	False	True
LoRA r	N/A	N/A	8	N/A	8
LoRA Alpha	N/A	N/A	64	N/A	64
LoRA Dropout	N/A	N/A	0.1	N/A	0.1
LoRA Target	N/A	N/A	*_proj	N/A	*_proj

Table 3: Detailed hyper-parameter settings for baselines and our method. By default, the parameters are for the Wiki-SS NQ training. If the setup is different for SlideVQA training, it is noted in parentheses.