Solution Set- Document AI

by **Shrey Mishra**

A practice assignment presented for a full time position at Alpha10x.

All code is available at:
https://github.com/mv96/alpha10x-assignment

Contact Information:
Mobile: (+33) 752-144-646
Email: mishra@di.ens.fr
LinkedIn: https://www.linkedin.com/in/shreymishramv96/
ENS Ulm Paris
PSL University
Pais

November 26, 2024

1 Introduction

We will enter the exploratory phase to understand our data and its differences or similarities to well-known tasks within the document AI community. Instead of building our own solutions from scratch, we can leverage existing ones. This will also aid us in identifying the problem statement more effectively.

1.1 Objective

Before beginning with any assignment it is almost always better to understand the problem statement and the nature of the problem. We are provided with 2 documents (both publicly available): a technical report from Deloitte *deloitte_en_lithium_pov_cn_20221114.pdf (View here)* which contains information about China's Lithium Industry, and another report *striking-gold-with-ev-battery-recycling (View here)* from BCG group about EV battery recycling.

1.2 Questions to Address

We need to address the following questions:

- What is the expected size of the battery recycling market in 2030?
- What are the two main techniques for extracting raw materials from end-of-life EV batteries?

1.3 Key Assumptions

Each question relates to a specific PDF. Our system cannot rank or assign relevant PDFs based on the question. Therefore, we will assume that the first question pertains to the Deloitte report and the second to the BCG report.

1.4 Baseline evaluation - `GPT-4o-mini`

We can also establish a pseudo ground truth by utilizing GPT models as a baseline for evaluation [21], which will serve as a reference point for comparison. For further insights, you can access the conversation interface here: `https://chatgpt.com/share/67407d00-c734-8000-9261-ee072ff2d948`

1.4.1 Deloitte Report

The global EV battery recycling market is projected to exceed $46.5 billion by 2030, driven by the rapid growth in electric vehicles and an increasing wave of battery retirements.

1.4.2 BCG Report

The two main techniques for extracting raw materials from end-of-life EV batteries are:

- **Pyrometallurgy:** A thermal-based extraction and purification process. It is robust and can handle a wide range of battery chemistries but typically recovers fewer recycled materials and requires high energy inputs.
- **Hydrometallurgy:** Uses aqueous chemistry to dissolve and separate metal components for extraction and purification. This method is more material-efficient and can achieve higher recovery rates but requires significant investment and advanced technology.

1.5 Challenges and Feasibility

We aim to build a Question-Answering (QA) system that takes a text query and a PDF document as inputs and provides answers based on the content of the PDF. However, before diving into development, it's crucial to address the challenges associated with this task and explore what can realistically be achieved within limited time and resources.

Here are few important points to consider:

1. **The information is visually rich and layout information is important:** This means that simple conversion and extraction of the text from the PDF is not good enough; often we need to look at graphs (visual components) to answer specific questions such as the information present in graphs. This relies on a mixed dependency on both modalities since there are numbers embedded inside the PDF that may or might not have been addressed in the text section due to the visually rich nature of the document. The layout information is also extremely important in such technical reports because they do not follow a rigid scientific structure to make it visually appealing for the readers. For example, the B part in the figure 1 indicates the recycle amount, whereas in a scientific paper, a graph would have clearly denoted the x-axis at the bottom of the graph. So even if this text is captured, we need to capture the right location, i.e., the layout part.

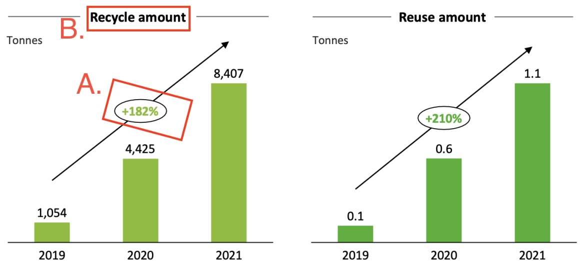

Figure 1: visually rich information present in graph A. denotes the growth B. denotes the varied style

2. **The non-scientific nature of the PDFs presents challenges for analysis:** These documents are not structured like scientific papers; for instance, author names are placed at the end, and they lack mathematical expressions or formulas. Additionally, they do not adhere to a strict formatting style, which complicates their use with specialized tools such as Grobid and pdfalto which are specialised for scientific mining and may have helped reduce the run time and avoid OCR solutions. Despite these challenges, I attempted to process the PDFs with the Grobid server [1] and obtained some interesting results (see figure 2).

   To a large extent, I was able to successfully recover segments such as text and images, as well as the key takeaways identified in `Deloitte.pdf` as abstracts by Grobid.

   However, preprocessing these elements for practical use is beyond the scope of this assignment. This limitation is due to the restricted time available and unavailability faced by state-of-the-art (SOTA) general long document QA systems like PDF-WuKong [28]. Nevertheless, this work could serve as a valid starting point for future research, if hired :)

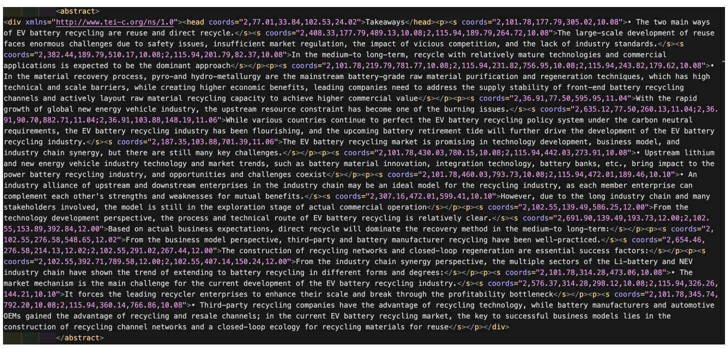

Figure 2: Grobid interpreting Key takeaways as abstract

3. **Long Document Nature:** Most of the well-established papers in the community of document AI are benchmarked on datasets such as FUNSD [13] and RVLCDIP [11], which mostly consist of short-page documents like bills, receipts, and memos. These types of documents are easier to handle by models such as LAYOUTLM (all versions) [30] [29] [12], NOUGAT [7], UDOP [26], LILT [27], Docformer [3], and Donut [16]. This means that we are operating on a strong asusmption that the answer to the question is bounded within the specifc page making our system linearly scale with number of pages inside the pdf. for reference, see figure 3 directly taken from the UDOP paper.

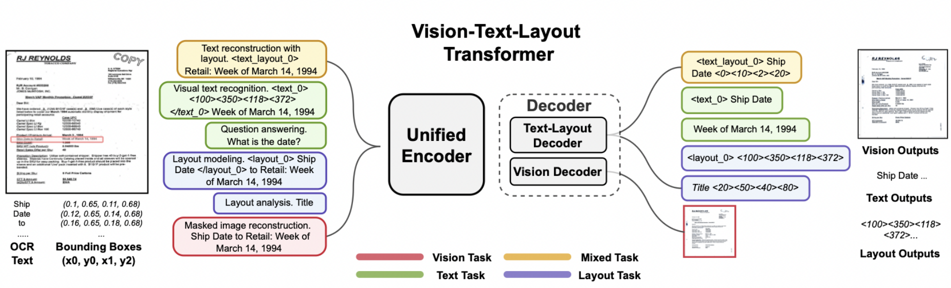

Figure 3: UDOP operating on a strong asusmption that the document is a one pager bill

2 Model inference

The ground truth answers are generated using **GPT-4**. While this method may not represent the optimal approach, it provides a reliable benchmark for evaluating the performance of our models. We will evaluate our models using a standard workflow, making incremental adjustments with each iteration to enhance the quality of our solution. Let's first understand the base solution outlined below:

To assess the performance and accuracy of the models, we employ two primary metrics: BERTScore [32] and ROUGE [18].

2.1 Short-OCR-Text-Models

1. **PDF Conversion**
   - **Objective**: Transform the PDF document into bitmap images [5].
   - **Method**: Utilize threading to parallelize the conversion process, thereby reducing the total processing time.
2. **Optical Character Recognition (OCR)**
   - **Objective**: Extract textual content from each bitmap image.
   - **Method**: Apply OCR [25] on the extracted images to obtain text output for every page in the PDF.
3. **Model Integration**
   - **Objective**: Load and utilize multiple pre-trained language models for question answering.
   - **Models Used**:
     - `twmkn9/distilbert-base-uncased-squad2` [24]
     - `deepset/bert-large-uncased-whole-word-masking-squad2`[8]
     - `deepset/roberta-base-squad2`[19]
     - `allenai/longformer-base-4096`[4]
   - **Considerations**:
     - These models vary in their number of parameters and context lengths.
     - Some models may not accommodate the entire text of a single page, influencing their performance and efficiency.
4. **Inference Process**
   - **Objective**: Answer user-provided questions based on the extracted text.
   - **Method**: Iterate through each page's text, passing it to the loaded models for inference.
   - **Performance**: The inference time scales linearly with the number of pages in the PDF, as each page is processed sequentially.
5. **Modality Limitation**
   - **Scope**: The models exclusively process textual data.
   - **Implication**: They can only answer questions that are directly addressed within the provided text, limiting their applicability to text-based queries.

2.1.1 Performance and Observations

Before we can utilize the machine learning model for inference, we must consider the preprocessing time. The conversion of the PDF to bitmap images takes approximately 3.18 seconds (see `pdf2img.py`), followed by the OCR process, which requires about 11.94 seconds (see `img2ocr.py`). [1]

Table 1: Model Performance Metrics for Short-OCR-Text-Models

Model Name	Inference Time (s)	BERTScore F1 (%)	ROUGE-L (%)	Model Size (M)
twmkn9/distilbert-base-uncased-squad2	3.7807	30.15	0.00	66.36
deepset/bert-large-uncased-whole-word-masking-squad2	**33.6416**	**35.42**	**6.25**	**334.09**
deepset/roberta-base-squad2	10.9366	31.50	0.00	124.06
allenai/longformer-base-4096	32.2392	32.28	6.06	148.07

`Best Model:` Bert Large-334M (trained on SQUAD2 [23])
`Total time:` 48.76 seconds (includes the preprocessing time)
`ROGUE-L:` 6.25 %
`BERTSCORE:` 35.42%
`Query:` *What is the expected size of the battery recycling market in 2030?*
`Best answer found on page 5:` *145 tonnes of lithium carbonate equivalent*
`Model confidence:` 0.53

The Short-OCR-Text Only Model provides a foundational understanding of how basic language models perform in question-answering tasks using OCR-extracted text from PDFs. While effective for simple queries, limitations in context handling and resource demands highlight the need for more advanced models or optimization strategies for larger and more complex documents.

2.2 Long-OCR-Text Models

In this section, we process the OCR-extracted text using models with extended context windows. The goal is to efficiently answer questions and reduce inference time by leveraging large context models, thereby minimizing the number of required forward passes. We utilize generative models such as the `Mistral-7B-Instruct` [14] and `Llama-3.2-1B-Instruct` [9] models, both of which offer extensive context windows that enable the retention of contextual information across multiple pages. Additionally, page delimiters are incorporated in the combined text files, as demonstrated in `combined_texts.py`

Our approach continues to employ OCR for converting images to bitmap format. Notably, the model size has significantly increased from a 334M-parameter model to a billion-parameter model, enhancing its capability to handle more complex queries.

We report the ROUGE-L and BERTScore metrics in Table 2, along with the number of parameters for each model.

Table 2: Model Performance Metrics for Long-OCR-Text Models

Model Name	Inference Time (s)	BERTScore F1 (%)	ROUGE-L (%)	Model Size (B)
mistral/7b	16.94	**57.58**	**18**	7
llama/3.2-1b	**16.26**	53.06	13.7	**1.3**

`Best Model:` llama-3.2-1b-Instruct
`Total time:` 31.38 seconds (includes the preprocessing time)
`ROGUE-L:` 13.71 %
`BERTSCORE:` 53.06%
`Query:`

Context: `<context form @combined_output.txt >` Question: *What is the expected size of the battery recycling market in 2030?* Please provide a concise answer based on the context above. Also, reference the pages at the beginning where you found the answer (if any). Additionally, include a confidence score between 0 and 1 indicating how sure you are about the answer.

`Answer:`

Based on the provided text, I was unable to find a specific mention of "the expected size of battery recycling in 2030. However, according to Deloitte's research report, "Electrification Trends: The Rise of Electric Vehicle Charging Infrastructure," which was referenced at page 5 of the provided text: "The market is expected to grow from $45 billion in 2020 to $140 billion by 2025 and reach $300 billion by

[1] All inference times are based on an M1 MacBook running locally and are subject to change.

2030." The confidence score for this answer is 0.8, indicating a moderate level of certainty based on the report's data. It is worth noting that Deloitte's research reports are typically based on extensive industry analysis and forecasts, but they may not always provide a specific estimate for a particular year or scenario.

References:

- "Electrification Trends: The Rise of Electric Vehicle Charging Infrastructure" (Deloitte, 2022)
- Page 5: "Market size projections"
- Confidence score: 0.8

`Model confidence:` 0.8

Running models on Long-OCR-text demonstrates the effectiveness of larger context windows. Despite the increase in model size to a billion parameters, the inference time remains lower because we only perform a single forward pass. However, the model's performance was not perfect, as it requires a multimodal approach to accurately interpret answers presented in visual graphs.

2.3 Short-OCR-Multimodal Models

In this section, we evaluate models that process one page at a time while utilizing multimodal inputs. Specifically, we use **LAYOUTLM**, which integrates layout information—such as text extracted via OCR along with their coordinates—and incorporates a full-page image to establish visual relationships.

We employ the latest version of LAYOUTLM, namely **LayoutLMv3**, which has been trained on the DocVQA task (see `rubentito/layoutlmv3-base-mpdocvqa`). The top two answers from this model predicts `63%` on page 6 and `2.14` on page 15 within the table. While the model effectively interprets charts and growth patterns, it struggles with metric interpretations as in the model fails to identify that the answer has to be in the form absolute monetary amount. This difficulty may stem from the model's relatively small size of 125 million parameters or potentially indicates the need for further fine-tuning.

Additionally, we explore other state-of-the-art vision-language models that accept input images and provide answers based on the given context and questions. Examples include **LLAVA**, **BAKLLAVA**, and the **Llama 3.2 11B multimodal model**.

Below is the table summarizing the results:

Table 3: Model Performance Metrics for Short-OCR-Multimodal Models

Model Name	Inference Time (s)	BERTScore F1 (%)	ROUGE-L (%)	Model Size (B)
llama/3.2-11b vision	126.52 (per page)	43.32	11.32	11
Layoutlmv3	68.026	33.61	0	0.125
LLAVA	12.438(per page)	**62.47**	**22.76**	7
BAKLLAVA	5.49 (per page)	41.58	12.12	7

`Best Model:` LLAVA
`Total time:` 437.74 seconds (includes the preprocessing time)
`ROGUE-L:` 22.76 %
`BERTSCORE:` 62.47%
`Query 1 (for Layoutlmv3):`

Context: `<context form @ocr_output.txt >` Question: *What is the expected size of the battery recycling market in 2030?* Please provide a concise answer based on the context above. Also, reference the pages at the beginning where you found the answer (if any). Additionally, include a confidence score between 0 and 1 indicating how sure you are about the answer.

`Query 2 (for all other generative moodels):`

OCR scanned text: `<context form @ocr_output.txt>`
Question: *What is the expected size of the battery recycling market in 2030?*
Please provide a concise and accurate answer to the question based on the provided OCR text and image.
`Answer:` Best answer found on page 6:

The image shows a chart titled G̈lobal EV battery recycling market size projection,ẅhich forecasts the market size for the global electric vehicle (EV) battery recycling industry, expected to reach $10 billion by 2030. This growth is attributed to increased efforts in EV battery recycling and recovery projects. The chart breaks down the size of the market into four categories: lithium, nickel, cobalt, and manganese. As of 2021, the lithium category holds the largest share within this industry. The chart is based on Anson Securities' analysis and includes a disclaimer regarding the estimates.

2.4 Short-NOOCR-Multimodal Models

In this section, we explore OCR-free models capable of performing question-answering tasks on single-page documents. Two notable examples in this category are the Donut model and the UDOP (Unified Document Processing) model.

The Donut model demonstrates proficiency in question-answering tasks, while UDOP offers enhanced capabilities. Beyond basic QA functionality, UDOP excels at processing layout-related queries on single-page documents within a multimodal context.

The Donut model answers `63%` when running inference on page 6 while the best confidence score is for page 26 where the model answers 10.5gwh. Similarly for UDOP model we have the answer `The expected size of the battery recycling market in 2030 is expected to exceed 10 billion dollars.` which is accurate depiction on the growth but not the precise market value. The model also beleives the best answer is on page 2 stating `The battery recycling market is expected to be the dominant approach in 2030`.

The answers from UDOP are faster and accurate toward the actual answers.

Table 4: Model Performance Metrics for Short-NOOCR-Multimodal Models

Model Name	Inference Time (s)	BERTScore F1 (%)	ROUGE-L (%)	Model Size (M)
naver-clova-ix/donut-base-finetuned-docvqa	326.98	44.36	18.52	200.42
microsoft/udop-large	132.4660	65.76	37.21	741.65

`Best Model:` udop-large
`Total time:` 135.58 seconds (does not include the OCR time)
`ROGUE-L:` 37.21 %
`BERTSCORE:` 65.76%
`Query:` What is the expected size of the battery recycling market in 2030?
`Answer:` Best answer found on page 2: The battery recycling market is expected to be the dominant approach in 2030.

2.5 LONG-NOOCR-Multimodal Models

If I had more time, I would explore models like PDFWuKong, which are capable of capturing long-context information. PDFWuKong builds upon the work of Text Monkey [20] and is similar to some of the models already explored in the NOOCR section. It also supports PDF parsers like Grobid, which can automatically segment PDFs into blocks of multimodal information without requiring OCR.

Currently, our model evaluates every page of the PDF, which results in significant training time being wasted. A simple solution would be to use Retrieval-Augmented Generation (RAG) to compute similarity scores and select the top-k PDF pages where the context is most relevant.

3 Conceptual questions

The Colpali model [10] is based on the ColBERT model [15] and the Paligemma model [6]. The primary task of the Colpali model is to provide an enhanced ranking system to identify relevant pages of a document for better contextualization. Colpali utilizes the multimodal embeddings generated from the Paligemma model. Paligemma employs the Seglip model [31] internally, which is faster to compute compared to the CLiP model [22]. Consequently, the embeddings offer superior representation along with much faster inference. This advantage is derived from the Paligemma model.

The ColBERT model [15] is then used to better relate the ranking of the query system to the relevant page or subset of pages through a late interaction mechanism. Thus, Colpali combines the best of both worlds; it effectively ranks documents based on the embeddings received from the Paligemma model.

Colpali eliminates a significant portion of the Optical Character Recognition (OCR) processing, which is a common first step in a typical Retrieval-Augmented Generation (RAG) system [17]. Moreover, standard RAG systems are not multimodal, whereas Colpali operates on scanned images. This capability allows for querying both graphs and text with the Colpali model.

3.0.1 How does ColPali work ?

Colpali processes a document image by passing it through the vision LLM, which in this case is PaliGemma. This model generates the textual representation of the decoder LLM. The idea is to use these embeddings in a vector store to perform a similarity search with the given user query when the system is online. The similarity search is conducted in a ColBERT fashion, taking the contextualized query embedding and comparing it with all the relevant token embeddings generated by the LLM.

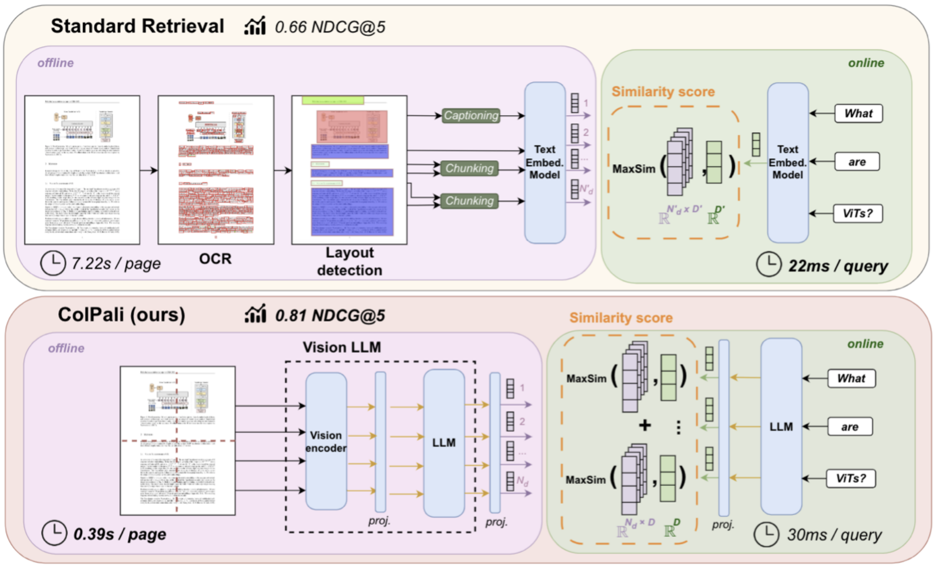

Figure 4: Colpali Model Architecture

3.0.2 Training methodology and use case

Colpali is ideal for training document retrieval tasks where multimodal retrieval is required. User queries can often specifically point to graphs, which can be challenging to capture and are the intended target for Colpali (see Figure 1 of the Colpali paper). It can also be used to find relevance among old papers that are handwritten or written in low-resource languages, as Paligemma supports many languages. We can fine-tune an already pretrained Paligemma to rank our specific documents, or we can simply use the Colpali-Gemma model already trained in a zero-shot setting.

The main advantage of Colpali is its ability to create rich textual embeddings by taking multimodal context into account and using the ColBERT ranking methodology. However, capturing very specific domain knowledge can be challenging.

There are major disadvantages to Colpali as well. The time saved on OCR is not significant compared to existing fast PDF parsers like Grobid (used in Semantic Scholar), which also generate PDF XML. While Grobid chunks do cover page breaks, Colpali may have trouble capturing information that is split across several pages. The performance of Colpali is also dependent on the base encoder models, i.e., how good they are at math, reasoning, or multilingual tasks compared to, say, Phi-3.5 [2] with the ColBERT late interaction mechanism.

Compared to my current system, Colpali is quite fast, but it is worth noting that encoding every single forward pass requires passing the data through a 3 billion parameter model. In my study, I have proposed models at various complexity levels that can be used and deployed according to the hardware and problem specifications. For very simple tasks such as QA in receipts, a simple LayoutLM or a text-only RoBERTa model could be low-resource. If I had time, I would have implemented the Colpali model and compared it with the standard RAG over some of the architectures I built.

Also, a minor remark: Colpali alone is designed mainly for document retrieval but can easily be extended to QA tasks by using the decoder model to generate some responses.

4 System Design Question

Query:

> Give me as complete as possible company profiles for Li-Cycle, Fortum, GEM, Umicore, Ganfeng Lithium, from the information provided in the documents.

- *Description*
- *Location*
- *Recycling technology used*
- *Business Model*

There are two major challenges in addressing the above query:

1. **Context Understanding and Retrieval:** This query requires descriptions and locations for different company profiles. If there are multiple company profiles, we need to ensure that each is used in the correct context and understand how they are related. Think of it like a graph; a plain extraction may not successfully build connections between the nodes, as it may extract the nodes but not the connections.

2. **Information Linking:** Even if we could extract the information, it is challenging to practically link all of this extraction in a single query (assuming the information is present in infographics and not on the same page). This means the query has to be broken into simpler subqueries, which will increase subsequent calls to the model. I would prefer using models that take the entire PDF as input and have a long context, either using models like PDFWuKong [28] or using PDF parsers such as Grobid and training another model on top of all the Grobid-extracted text.

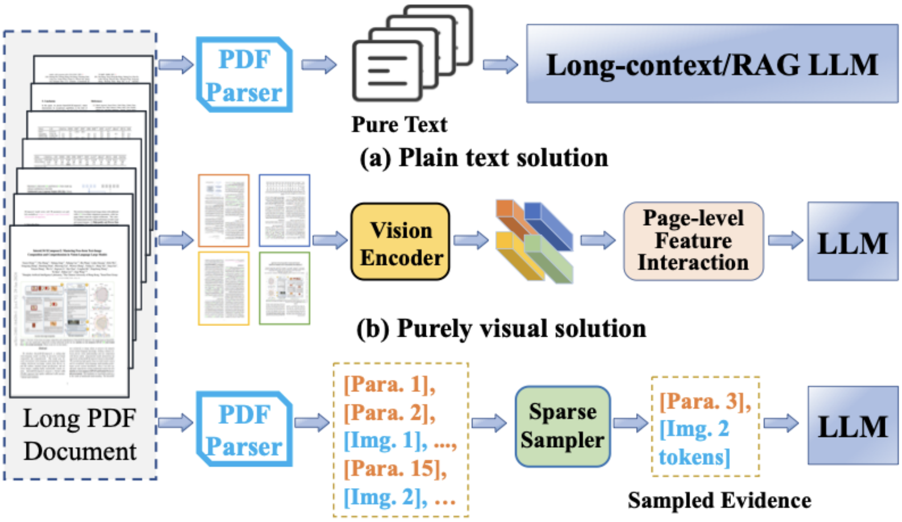

Figure 5: PDF-Wukong Model Architecture

If I had time, I would use Grobid or PDFAlto to extract the entire PDF into valid XML, because this process is faster than converting bitmap images and logically parses the PDF into sentences and paragraphs, while also identifying scientific symbols and fonts. From the multimodal paragraph extraction, I would embed the information of every paragraph into a multimodal embedding space. Passing this information through a transformer layer can help capture long dependencies that may occur several pages apart. There are many long attention mechanisms well-suited for this task. We could even switch to Mamba with linear time complexity since attention scales in order of n^2, or we can replace attention with FlashAttention2 for faster inference. The idea is to connect several paragraphs together by adding another transformer layer on top of the existing multimodal embeddings to capture context. We can then build a RAG on the contextualized multimodal embedding that can be used with a RAG to perform the QA task.

The only major drawback of such a system would be that it won't work on handwritten documents since Grobid will fail to extract information from non-machine-written documents. In such a case, I propose converting text into a markdown or plain text format using either Colpali or any other OCR/OCR-free model such as NOUGAT.

This approach will ensure that for most digitally written documents, I can process them faster and with ease.

References

[1] Grobid. `https://github.com/kermitt2/grobid`, 2008–2024.

[2] Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat S. Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, 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, Chen Liang, Weishung Liu, Eric Lin, Zeqi Lin, Piyush Madan, 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, Xia Song, Masahiro Tanaka, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. *CoRR*, abs/2404.14219, 2024.

[3] Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. Docformer: End-to-end transformer for document understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 993–1003. IEEE, 2021.

[4] Iz Beltagy, Matthew E. Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

[5] Edouard Belval. pdf2image: A python module that wraps pdftoppm and pdftocairo to convert pdf to a pil image object, 2018. Accessed: 2024-11-24.

[6] Lucas Beyer, Andreas Steiner, André Susano Pinto, Alexander Kolesnikov, Xiao Wang, Daniel Salz, Maxim Neumann, Ibrahim Alabdulmohsin, Michael Tschannen, Emanuele Bugliarello, Thomas Unterthiner, Daniel Keysers, Skanda Koppula, Fangyu Liu, Adam Grycner, Alexey A. Gritsenko, Neil Houlsby, Manoj Kumar, Keran Rong, Julian Eisenschlos, Rishabh Kabra, Matthias Bauer, Matko Bosnjak, Xi Chen, Matthias Minderer, Paul Voigtlaender, Ioana Bica, Ivana Balazevic, Joan Puigcerver, Pinelopi Papalampidi, Olivier J. Hénaff, Xi Xiong, Radu Soricut, Jeremiah Harmsen, and Xiaohua Zhai. Paligemma: A versatile 3b VLM for transfer. *CoRR*, abs/2407.07726, 2024.

[7] Lukas Blecher, Guillem Cucurull, Thomas Scialom, and Robert Stojnic. Nougat: Neural optical understanding for academic documents. *arXiv preprint arXiv:2308.13418*, 2023.

[8] 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: Human Language Technologies*, pages 4171–4186. Association for Computational Linguistics, 2019.

[9] Abhimanyu Dubey and 532 others. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[10] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. *CoRR*, abs/2407.01449, 2024.

[11] Adam W. Harley, Alex Ufkes, and Konstantinos G. Derpanis. Evaluation of deep convolutional nets for document image classification and retrieval. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 991–995. IEEE, 2015.

[12] Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. Layoutlmv3: Pre-training for document ai with unified text and image masking. *arXiv preprint arXiv:2204.08387*, 2022.

[13] Guillaume Jaume, Hazım Kemal Ekenel, and Jean-Philippe Thiran. FUNSD: A dataset for form understanding in noisy scanned documents. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 2, pages 1–6. IEEE, 2019.

[14] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

[15] Omar Khattab and Matei Zaharia. Colbert: Efficient and effective passage search via contextualized late interaction over BERT. In Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu, editors, *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pages 39–48. ACM, 2020.

[16] Geewook Kim, Teakgyu Hong, Moonbin Yim, Jeongyeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. Donut: Document understanding transformer without ocr. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 322–339. Springer, 2022.

[17] Patrick S. H. Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

[18] Chin-Yew Lin and Franz Josef Och. Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 605–612, Barcelona, Spain, 2004. Association for Computational Linguistics.

[19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.

[20] Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. Textmonkey: An ocr-free large multimodal model for understanding document. *CoRR*, abs/2403.04473, 2024.

[21] OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence, 2024. Accessed: 2024-11-24.

[22] 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. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 2021.

[23] Pranav Rajpurkar, Robin Jia, and Percy Liang. Know what you don't know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789. Association for Computational Linguistics, 2018.

[24] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *Proceedings of the 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing*, 2019.

[25] Ray Smith. An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*, pages 629–633. IEEE Computer Society, 2007.

[26] Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. Unifying vision, text, and layout for universal document processing. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pages 19254–19264. IEEE, 2023.

[27] Wenwen Wang, Yiheng Xu, Tengchao Lv, Lei Cui, Min Lin, and Furu Wei. Lilt: A simple yet effective language-independent layout transformer for structured document understanding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2138–2148. Association for Computational Linguistics, 2022.

[28] Xudong Xie, Liang Yin, Hao Yan, Yang Liu, Jing Ding, Minghui Liao, Yuliang Liu, Wei Chen, and Xiang Bai. Pdf-wukong: A large multimodal model for efficient long PDF reading with end-to-end sparse sampling. *CoRR*, abs/2410.05970, 2024.

[29] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. Layoutlmv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, pages 2579–2591. Association for Computational Linguistics, 2021.

[30] Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 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*, pages 1192–1200. Association for Computing Machinery, 2020.

[31] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 11941–11952. IEEE, 2023.

[32] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with BERT. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.