A-SEA³L-QA: A Fully Automated Self-Evolving, Adversarial Workflow for Arabic Long-Context Question-Answer Generation

Kesen Wang
Humain
kwang@humain.ai

Daulet Toibazar
Humain
dtoibazar@humain.ai

Pedro J. Moreno
Humain
pmoreno@humain.ai

Abstract

We present an end-to-end, self-evolving adversarial workflow for long-context Question-Answer (QA) Generation in Arabic. By orchestrating multiple specialized LVLMs: a question generator, an evaluator, and a swarm of answer generators, our system iteratively refines its own performance without any human intervention. Starting from raw, multi-page Arabic documents across diverse domains, the question generator produces fine-grained, context-aware queries to be tackled by the answer generator swarm, and the evaluator assesses and feeds back quality metrics. This closed-loop cycle enables continuous learning: low-confidence outputs trigger automated re-generation and model updates, progressively enhancing question difficulty and relevance. Moreover, we set the quality metrics as a tunable hyperparameter, enabling question generation at controllable and customizable difficulty levels. We release ***AraLongBench***, a large-scale Arabic benchmark of single- and multi-page challenges spanning hundreds of pages, and demonstrate that our self-evolving workflow substantially outperform static pipelines, markedly boosting the long-context comprehension capabilities of leading Arabic Large Vision Language Models (LVLMs). Lastly, we also meticulously architect a fully automated agentic workflow for long-context Arabic document collection. [1]

1 Introduction

Document understanding (DU) in vision-language research remains an essential yet challenging issue, particularly for documents with complex layouts and lengthy contextual dependencies. Over the past few years, large vision-language models (LVLMs) have achieved remarkable progress on short-context tasks involving documents. Closed-source LVLMs such as OpenAI's *GPT* series (Achiam et al., 2023; OpenAI, 2024b,a), Google's *Gemini* (Gemini Team, 2024), and Anthropic's *Claude* series (Anthropic, 2024), and open-source models such as *InternLM-XC2-4KHD* (Dong et al., 2024), *LLaVA-NeXT* (Li et al., 2024), and *CogVLM* (Wang et al., 2023) have all demonstrated strong performance in comprehension of documents with complex layouts when there is limited context length. The models excel on single-page visual question-answering and reasoning benchmarks, such as *DocVQA*, *ChartQA*, and *InfographicVQA*, as well as other associated datasets (Mathew et al., 2021; Masry et al., 2022; Mathew et al., 2022; Zhu et al., 2022). This achievement showcases the promise of LVLMs for DU tasks when there is a limited context length.

However, current LVLMs struggle to generalize their success to long-context DU tasks involving multi-page documents and long-range reasoning (Xu et al., 2023). On challenging multi-page question-answering benchmarks (e.g. *MMLongBench*, *LongDocURL*, *M-LongDoc*), even the best LVLMs reach only about 40% accuracy, and many perform worse than text-only LLM baselines that rely on OCR-extracted text (Ma et al., 2024; Deng et al., 2024; Chia et al., 2024). This shortfall highlights the difficulty LVLMs have in capturing long-range and cross-page dependencies. A primary reason is the lack of training data with diverse, fine-grained questions whose answers are distributed across multiple pages. This data scarcity is even more pronounced for low-resource languages like Arabic.

Up until now, the primary Arabic DU benchmark, *Camel* (Ghaboura et al., 2024) and *KITAB* (Heakl et al., 2025), focuses on single-page question answering over short passages and reports suboptimal accuracy for state-of-the-art models, highlighting both the scarcity of fine-grained Arabic QA data and the high error rate of existing pipelines. These limitations prevent LVLMs from capturing

---

[1] https://github.com/wangk0b/Self_Improving_ARA_LONG_Doc.git

long-range dependencies or cross-page semantics in Arabic documents. To overcome these gaps, we propose a self-evolving, multi-LVLM collaborative workflow: autonomous layout-parsing, question-generation, and evaluation workflow that iteratively enhance knowledge depth, enrich question diversity, and refine Arabic long-document QA without human intervention, culminating in a large-scale, multi-page Arabic QA generation pipeline. In sum-
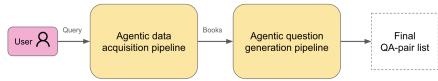

Figure 1: High-level Abstract of the Automated Pipeline

mary, our key contributions are as follows:

1) Addressing Arabics low-resource challenges: We design and deploy an autonomous data-collection agent to aggregate extensive and long-context Arabic corpora.

2) Fully automated, self-evolving adversarial question generation for Arabic documents: we propose a closed-loop, automated workflow comprising layout parsing, question generation, and quality evaluation LVLMs that iteratively refine their outputs to produce high-quality multi-page Arabic QA pairs across diverse domains from long, raw documents with only a single prompt.

3) Rigorous evaluation on Arabic LVLMs: We curate **AraLongBench**, a large-scale, multi-page Arabic QA benchmark, and perform extensive zero-shot evaluations with leading Arabic LVLMs. Results show that our generated data significantly exposes persistent weaknesses in the major LVLMs when it comes to Arabic long-context DU, guiding future model improvements.

Collectively, these contributions advance long-context Arabic DU by delivering an end-to-end, self-evolving adversarial workflow for data annotation (Figure 1 presents a high-level abstract of the entire workflow), a publicly available benchmark, and a fully automated Arabic data acquisition pipeline, laying the groundwork for the training of more robust real-world LVLMs in long-context Arabic DU.

2 Related Work

2.1 Arabic DU Datasets

A number of datasets have been developed to facilitate document understanding for various tasks, and a growing body of work has begun to address Arabic documents. There are also early Arabic layout analysis benchmarks like *BCE-Arabic-v1* (Saad et al., 2016), which brings together 1,833 scanned pages of 180 books with various fonts, multi-column layouts, photos, tables, and charts, as a benchmark for DLA, OCR, and text-to-speech research; and *BADAM* (Kiessling et al., 2019), a 400-annotated manuscript image dataset spanning historical and contemporary domains, to serve as a baseline detection benchmark in Arabic-script documents. More recent efforts have produced larger and more diverse sets. For instance, *SARD* (Nacar et al., 2025) offers 843,622 synthetically created book-like images in ten Arabic fonts to offer typographic coverage and clean layouts, while *KITAB-Bench* (Heakl et al., 2025) is made up of 8,809 real-world instances in nine domains and 36 sub-domains (including tables, charts, and mixed handwritten/printed text) to evaluate modern OCR and DU methods.

Despite these advances, existing Arabic datasets remain largely restricted to single pages (scanned or artificial) and limited domains, which limits their ability to test models on long-context tasks such as cross-page co-reference, layout changes, and heavily interleaved content. To bridge this gap and enable strict testing and training of multilingual LVLMs on truly long-document Arabic material, we must develop a large-scale, multi-page Arabic DU benchmark that combines real-world diversity (books, reports, manuals, and web archives), fine-grained annotations for layout elements, tables, figures, and cross-page structures, and automatically generated tasks covering summarization, information extraction, VQA, and reasoning. Such a dataset would open the door to the next generation of Arabic-capable LVLMs and genuinely end-to-end long-context document understanding.

2.2 Vision-LLMs

DU models can be broadly categorized into two groups:

1. **Cascaded Approach**: These pipelines first apply an Optical Character Recognition (OCR) engine and then encode textual and visual features separately. Recent Arabic-focused examples in-

clude *Arabic-Nougat*, which finetunes vision transformers to convert book pages into structured Markdown, handling multi-column layouts and diverse fonts (Rashad, 2024). Another example is *Qalam*, a SwinV2-encoder + RoBERTa-decoder multi-modal LLM trained on over 4.5 million manuscript images, achieving under 1.2% WER on printed Arabic and 0.8% on handwriting (Bhatia et al., 2024).

2. **End-to-End Vision-Based Approach**: These models ingest raw document images and directly output text or structured representations, often via a unified transformer. Key Arabic and multilingual advances include *GOT (OCR-2.0)* (Wei et al., 2024a), a 580 M-parameter end-to-end model supporting slice- and whole-page inputs with long-context decoding. Another notable example is *QARI-OCR*, which adapts Qwen2-VL to Arabic using massive synthetic data, achieving state-of-the-art CER 0.061 and robust layout handling (Wei et al., 2024b).

Evaluations on *KITAB-Bench* show that LVLMs (e.g., GPT-4o, Gemini-2.0-Flash, Qwen, AIN) outperform classic OCR by nearly 50% in CER (Heakl et al., 2025) yet still struggle with multi-page reasoning and cross-page dependencies. In other words, their ability to capture *long document* phenomena, such as cross-page co-reference, evolving layouts, and dense interleaving of text, tables, and figures, remains under-explored. Robust evaluation on true long-context Arabic corpora is, therefore, a critical next step.

2.3 Automated Data Annotation Systems

Training LLMs or LVLMs at scale needs trillions of high-quality, well-annotated data points, which is out of human-alone annotation. Current annotation systems tend to employ autonomous AI agents for synthesizing and validating labels with less human engagement. *LabelLerr's* pipeline manages self-correction and active-learning loops to label millions of images on its own, realizing a reduction in manual effort of over 50% with accuracy over 90% (Labellerr Inc., 2024). *LandingAIs* agentic document extraction uses vision-language agents to detect form fields, tables, and checkboxes and to generate structured schemas end-to-end, without human intervention (LandingAI, 2025). In the Arabic domain, *Arabic.AIs* ecosystem enables template-driven report generation but still requires manual setup and is not tailored for raw document annotation tasks (Tarjama (Arabic.AI), 2025); likewise, *UiPaths* Active Learning DU pipeline in-

corporates human-in-the-loop guidance but offers limited support for right-to-left scripts and complex multi-column layouts (UiPath, 2025). To our knowledge, no such system fully automates long-context Arabic DU annotation, demonstrating the novelty and timeliness of our fully automated multi-LVLM interactive workflow.

3 Fully Automated Workflow for Data Collection

We constructed our long document Arabic corpus by automatic web crawling of a number of online repositories with a multi-stage filtering and normalization pipeline for breadth and fidelity. We initially discarded pages with fewer than the minimum characters, pages under restrictive licenses, and documents that are not suitable for automated QA generation. HTML content was extracted with a DOM clever scraper built on *BeautifulSoup* (Richardson, 2007), and native PDFs were handled by *pdfplumber* to maintain layout and pull out text blocks (Smiley, 2020). Scanned paper documents and images were read with *Tesseract OCR* (Smith, 2007) with custom preprocessing (binarization, deskewing) to maximize legibility.

There were Arabic-specific problems that required additional steps. Right-to-left directionality and mixed Unicode encoding produced character misalignment, and we added a bidirectional-text handler based on the Unicode Bidirectional Algorithm (Unicode Consortium, 1996).

With these unified preprocessing efforts, our dataset realizes multi-page coherence and varied layout coverage, laying a solid foundation for long document comprehension. In addition, the collected data spans across a variety of domains such as education, finance, governmental reports, news, social media, technical manuals, etc.

Figure 2 illustrates an end-to-end automated, LVLM-controlled process to build an Arabic long-document corpus. The process involves four pri-
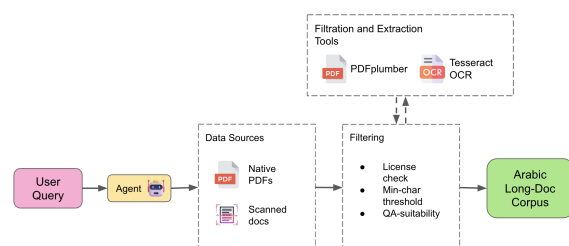
Figure 2: Automated Data Acquisition Workflow

mary stages:

1. **Agentic Query Dispatch:** For a high-level user request, an autonomous agent unit manages the following harvesting procedure, selecting appropriate repositories and search targets based on query semantics.

2. **Multi-modal Data Ingestion:** The agent retrieves candidate documents from diverse sources:

   - *Native PDFs:* Digitally created PDF documents downloaded from APIs or direct download.
   - *Scanned Documents:* Image-based documents (e.g., TIFF, JPEG) that require OCR to extract the text.

3. **Filtration and Extraction:** Raw inputs are processed by a modular toolset:

   - `PDFPlumber` to extract text and layout from native PDFs.
   - `Tesseract OCR` to recognize scanned images as machine-readable text (accuracy is not a major concern at this stage) that enables character counts.

   These components interact with the ingestion layer to support bidirectional refinement (e.g., re-crawling pages when layout anomalies are encountered).

4. **Quality-Controlled Filtering:** Automated screening of extracted documents is applied:

   - *License Compliance:* Checking against allowed reuse policies.
   - *Minimum Content Threshold:* Applying a character-count minimum to avoid evisceratingly brief texts.
   - *QA-Suitability Screening:* LVLM as a judge evaluation of each document's suitability for question-answer generation. To this end, we perform the filtering on a page-level with the document accepted as "QA-Suitable" only if $\geq 80\%$ of the pages pass the screening.

   Documents that satisfy all the criteria are aggregated into the final *Arabic Long-Doc Corpus*, facilitating downstream tasks such as structured question generation and large-scale language modeling.

4 Self-Evolving Adversarial QA Generator

4.1 Document Preprocessing

The preprocessing phase transforms raw PDF inputs into structured representations suitable for downstream tasks, following these key steps:

- **PDF Ingestion:** The framework accepts documents in PDF format.

- **Page Rasterization to Images** (I)**:** PDF pages are converted into image format using `pdf2image` to maintain original visual layout and contextual details (Belval, 2018).

- **Structural Layout Analysis** (L)**:** A deep-learning model (e.g., *DocLayout-YOLO*; (Zhao et al., 2024)) segments pages into logical elements such as headings, paragraphs, tables, and figures, enabling targeted content processing.

- **Document Chunking with Overlap** (I_c, L_c)**:** In order to process long documents in an efficient manner, pages are segmented into overlapping chunks with length 50-page and 5-page overlap. It yields segmented images I_c and structural layout annotations L_c for each chunk. Structural chunking was avoided due to the computational expense of page-level object detection and ordering, as well as the lack of availability of structural cues for scanned or poorly formatted documents. Fixed-size overlapping chunking was therefore selected for stability, scalability, and insensitivity to format variation.

4.2 Self-Evolving Adversarial Workflow

Following preprocessing, the multi-LVLM interactive workflow iteratively refines question-answer
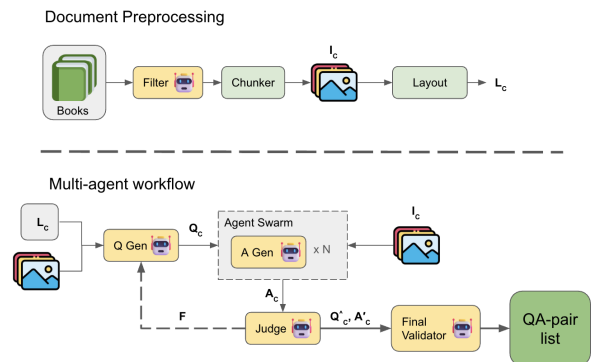

Figure 3: Self-evolving Question Generation Workflow

generation through the following structured sequence:

Q Gen: Question Generation

- *Input:* Image(s) and layout annotations (I_c, L_c).

- *Output:* Draft question Q_c (questions + cognitive premises), generated according to policy π, ensuring relevance and traceability to source content.

Agent Swarm: Answer Generation

- *Input:* Image(s) I_c and proposed question Q_c.

- *Output:* Candidate answers $\{A_{c_i}\}_{i=1,...,N}$ (answers + logical foundations), where N is the number of agents in the swarm, grounded explicitly in document content.

Judge: Assessment and Feedback

- *Input:* Full context data (I_c, L_c), questions Q_c, candidate answers $\{A_{c_i}\}_{i=1,...,N}$, and generation policy π.

- *Output:* Validated answers $\{A'_{c_i}\}_{i=1,...,N}$, question difficulty ratings, and actionable feedback F (correct answer + attempted answer + evaluation + suggested refinement) for question improvement.

Q Gen: Iterative Question Refinement

- *Input:* Feedback F from **Judge**.

- *Output:* Further refined question $\hat{Q}_c$, iteratively cycling through Step 1 until the desired quality and consistency are achieved.

Final Validator: Evidence Validation

- *Input:* Comprehensive context data (I_c, L_c), proposed question $\hat{Q}_c$, and validated answers A'_c.

- *Output:* Finalized questions $\hat{Q}_c$, each paired with rigorously validated evidence and answers.

Global Document Iteration: the iterative loop described above is repeatedly executed on every segment of the document, establishing a complete, verified collection of question-answer pairs for the entire document.

This multi-LVLM and collaborative method through repeated refinement ensures contextual correctness and robust validation, making Long DU an effective instrument for large-scale and intricate document understanding tasks. Detailed illustrations of the workflow is documented in Figure 3.

The workflow architecture utilizes prompted structured questions to sequence LVLM interaction on every step:

- **Question Generation Prompt (QGP):** Telling **Q Gen** to create detailed and reflective questions at three levels of complexity:

  – **Level 1 (Factual):** Questions requesting explicit information extraction from the text.
  – **Level 2 (Inferential):** Questions requesting logical reasoning and inference based on contextual clues.
  – **Level 3 (Contextual Ambiguity):** Questions that are context-derived but explicitly unanswerable from the provided document.

- **Question Refinement Prompt (QRP):** Guiding **Q Gen** to refine and improve the depth of its proposed questions based on the comprehensive feedback returned from **Judge**.

- **Answer Generation Prompt (AGP):** Instructing the **Agent Swarm** to produce accurate, contextually appropriate, and well-supported answers.

- **Assessment Prompt (AP):** Instructing **Judge** to evaluate question complexity, rejecting overly simplistic questions, and triggering iterative refinements towards improved quality.

- **Evidence Validation Prompt (EVP):** Commanding **Final Validator** to validate the source (e.g., tables, text, charts, etc) of the answers returned from **Judge**.

4.3 Iterative Refinement and Validation

With repeated cycles of iterative multi-LVLM cooperation, questions persistently evolve to maximize coverage, depth, and relevance:

- If **Judge** observes a greater than 50% accuracy rate in some question, it notifies **Q Gen** to raise question complexity, thereby challenging the **Agent Swarm** to elevate the difficulty level of the proposed questions.

- **Final Validator** strictly checks last question-answer pairs against verified sources, basically resolving contradictions and enhancing congruence against former observed benchmarks (Ma et al., 2024).

111

5 Data Analysis

From the initial pool of 1,301 Arabic candidate documents, we have retained 113 after subjecting them to a multi-stage filtering pipeline with an ending acceptance rate of 8.6%. The retained corpus spans a large number of domains including Legal (14), Medical (12), Research (10), Finance (10), Policy (9), Education (9), Manuals (8), News (8), Literature (8), Business (7), Technology (7), Environment (6), and History (5). Notably, OCR accuracy was not one of the most important issues in the recruitment process; OCR was employed solely as a surrogate to estimate character frequency and to verify that documents held a minimum of content.

The final dataset consists of well-structured tuples containing (question, answer, evidence pages, evidence sources, justification, and validation), making it a robust resource for long-document understanding research. Figure 4 presents the his-
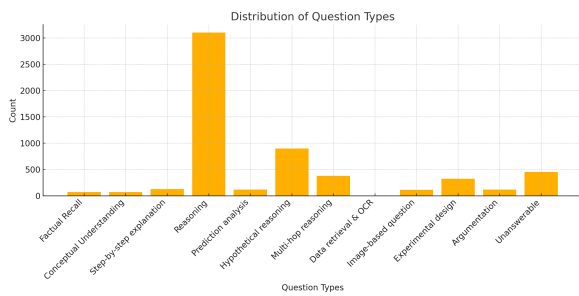

Figure 4: Proposed Question Types (50% Accuracy Threshold)

togram of 5,778 questions divided into twelve categories. Below is a detailed breakdown and interpretation.

Core & Hypothetical Reasoning (75.9%)

– **Reasoning:** 3,103 (53.7%) Emphasizes logical deduction, inference, and problem-solving skills.

– **Hypothetical Reasoning:** 898 (15.5%) Probes what-if scenarios, testing flexible application of knowledge under counterfactual conditions.

– **Multi-hop Reasoning:** 381 (6.6%) Chains together multiple inference steps for deeper, integrative understanding.

Integrity Checks & Multi-evidence Source Comprehension (9.9%)

– **Unanswerable:** 454 (7.9%) Assesses the ability to withhold answers when no valid solution exists, reducing hallucinations.

– **Image-based Question:** 112 (1.9%) Requires visual interpretation of charts, diagrams, or photographs.

– **Data Retrieval & OCR:** 6 (0.1%) Targets extraction of embedded or scanned text from documents.

Intermediate-Complexity Tasks (9.6%)

– **Experimental Design:** 321 (5.6%) Involves planning or critiquing scientific studies.

– **Prediction Analysis:** 118 (2.0%) Requires forecasting outcomes based on provided data.

– **Argumentation:** 116 (2.0%) Focuses on constructing or evaluating persuasive arguments.

Basic Comprehension & Procedural Explanation (4.6%)

– **Step-by-step Explanation:** 126 (2.2%) Demands clear, ordered procedural breakdowns.

– **Conceptual Understanding:** 72 (1.2%) Probes grasp of underlying principles rather than surface details.

– **Factual Recall:** 71 (1.2%) Tests straightforward retrieval of explicit information.

The dataset is heavily skewed toward reasoning (combined 75.9%), which are typically the most challenging tasks that require intensive and profound thinking. The second most challenging group of tasks including integrity checks (unanswerable), multi-modal questions where the answers are based on numerous sources (e.g., tables, charts, images, etc), and OCR represent the second largest population (9.9%) in the dataset. Tasks of intermediate complexity (9.6%), covering multi-step inference, experimental planning, and argumentation, are the second largest population in the generated dataset. Basic and simple comprehension and procedural items such as step-by-step explanation, conceptual understanding, and factual recall are the minority (4.6%).

6 Ablation Test

In this section, we conduct an ablation test on the relationship between the accuracy threshold and the distribution of the proposed question types. In

addition, we also verify that adding structural layout analysis increase the number of multi-modal questions.
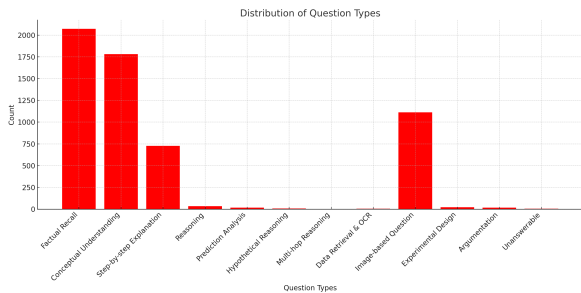
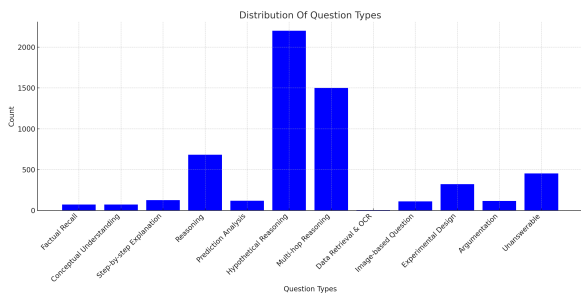

Figure 5: Proposed Question Types (No Accuracy Threshold)

Figure 6: Proposed Question Types (25% Accuracy Threshold)

By juxtaposing the original distribution (Figure 4), in which pure "Reasoning" questions dominated, with the low-threshold redistribution (Figure 5), we observe a striking re-balancing toward the hardest items. In fact, Figure 6 reveals "Hypothetical Reasoning" swelling to around 38% and "Multi-hop Reasoning" to around 26%, with "Reasoning" itself now below 12%. At the same time, elementary tasks like factual recall and conceptual checks shrink to under 5%, and mid-level formats (image questions, experimental design, argumentation) occupy only modest niches, while integrity checks (unanswerable/OCR) remain in place. In stark contrast, under extreme circumstances (without accuracy gate), Figure 5 shows nearly two-thirds of questions as simple deductive inference and fewer than 15% allocated to hypothetical or multi-hop chains. This side-by-side comparison confirms that relaxing accuracy constraints sharply redirects the generators output from straightforward inference toward the most complex, integrative reasoning challenges, ideal for stress-testing advanced models.

We apply *DocLayout-YOLO* to conduct structural layout analysis and retrieve bounding boxes
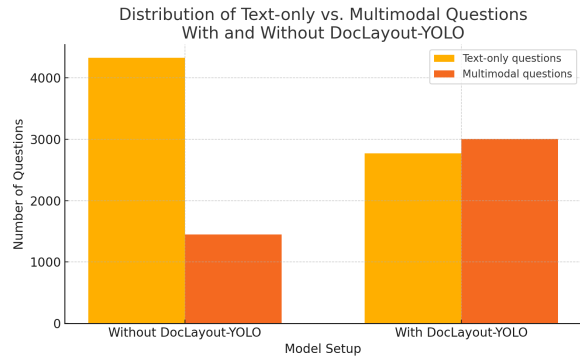

Figure 7: Distribution Of Text-Only Vs. Multi-modal Questions With and Without *DocLayout-YOLO*

of multi-modal elements such as tables, figures, charts, and other non-textual content. These regions are cropped and re-placed onto their corresponding document pages to form enriched page-level representations. Such a compositional strategy not only enhances multi-modal fidelity of information but also mitigates textual domination typically encountered in document processing pipelines, leading to a more semantically heterogeneous and balanced input space for downstream processing.

Figure 7 illustrates how the introduction of *DocLayout-YOLO* turns the rate of text-only questions against multi-modal questions. Under the "Without *DocLayout-YOLO*" mode, the system generated 4,327 text-only questions, nearly 75% of the output, against 1,451 samples (25%) that relied on visual content. When *DocLayout-YOLO* is introduced, however, the composition turns around: text-only questions fall to 2,771 (48%), and multi-modal questions rise more than twice to 3,007 (52%). This shift is something more than a statistical anomaly; it reflects a fundamental change in the preoccupation of the system. By accurately detecting and leveraging document layout elements (tables, figures, diagrams), *DocLayout-YOLO* unlocks a rich seam of visually grounded queries that were previously under-exploited. The increased use of multi-modal items not only provides diversity and complexity to the item pool but also forces downstream models to have to join text and graphics, which matters a lot. The histogram shows that adding layout awareness created a shift in generator focus that first made the multi-modal question type less than 25 percent; now it is a majority.

Lastly, we also involved human efforts in verifying the practicality of the **Final Validator**. First of all, we removed the **Final Validator** from the

Model	Param	CW	No Gate					50% Threshold					25% Threshold				
			SC	MC	LC	SP	CP	SC	MC	LC	SP	CP	SC	MC	LC	SP	CP
Closed-Source Models																	
GPT-4o	–	128K	87.2%	84.5%	83.1%	90.9%	82.8%	79.1%	78.2%	77.6%	83.5%	76.3%	65.7%	61.5%	59.8%	71.2%	64.5%
Gemini-2.0 Flash	–	1M	**93.0%**	87.2%	85.5%	**94.3%**	86.7%	**84.1%**	79.3%	79.0%	**85.2%**	80.1%	71.8%	68.2%	67.5%	73.9%	**72.0%**
Gemini-1.5 Pro	–	2M	90.0%	86.3%	79.3%	91.2%	88.4%	82.1%	79.0%	72.0%	83.0%	80.1%	**73.5%**	63.9%	64.5%	68.3%	67.6%
Gemini-2.5 Pro	–	2M	91.5%	**89.0%**	**88.2%**	93.4%	**90.2%**	81.7%	**80.1%**	**79.5%**	84.0%	**81.3%**	70.2%	**71.0%**	**68.0%**	**75.3%**	70.0%
Open-Source Models																	
AIN	7B	32K	78.5%	71.5%	67.7%	80.0%	71.5%	69.1%	62.3%	60.2%	71.0%	62.0%	58.2%	52.1%	49.3%	60.1%	50.6%
Aya Vision	32B	16K	79.1%	70.2%	66.7%	78.6%	70.4%	68.7%	61.0%	58.0%	71.2%	60.2%	57.3%	49.8%	46.9%	58.1%	50.0%
Qwen 2 VL	72B	32K	88.5%	84.0%	82.0%	90.0%	83.5%	78.7%	**75.1%**	**73.4%**	80.9%	74.0%	68.4%	63.0%	61.2%	71.0%	63.9%
Qwen 2.5 VL	72B	128K	**89.8%**	**85.2%**	**83.0%**	**91.5%**	**85.7%**	**79.6%**	74.8%	73.0%	**82.0%**	**74.5%**	**69.4%**	**64.8%**	**63.0%**	**71.5%**	**65.1%**

Table 1: Combined performance of LVLMs on the **AraLongBench** across three accuracy conditions (No Gate, 50%, 25%) and varying context lengths and page conditions. CW = context window; SC = short-context; MC = medium-context; LC = long-context; SP = single-page; CP = cross-page.

workflow and generated data as usual. Then, we randomly sampled 100 questions to inspect the reported evidence sources and found a 14% mismatch rate between the evidence source and the associated answer. The same process with this component back in the workflow was able to reduce the mismatch rate to below 5%.

7 Experiments

Table 1 provides an extensive comparative analysis of the performance exhibited by open-source and closed-source LVLMs on a variety of accuracy levels on the newly developed Arabic benchmarks in this work with differing accuracy thresholds. The benchmarks test the models under a wide range of context settings, categorically distinguished as short-context (SC: < 100 pages), medium-context (MC: 100-200 pages), and long-context (LC: > 200 pages), and tests based on single-page (SP) and cross-page (CP) conditions. Results are expressed as percentage accuracy to permit detailed observations on each model's performance in relation to the complexity of the task and linguistic intricacies inherent in Arabic.

Monotonically decreasing trends in performance are observed as we increase the accuracy bar across all models. On "No Gate", Gemini series and GPT-4o both get high-80s to low-90s across all page states and context lengths, demonstrating their true potential by being generously forgiving. A 50% gate threshold eliminates predictions on the margin, and mean scores decrease by approximately 10-12 points (Gemini-2.0 Flash SC from 93.0% to 84.1%; GPT-4o SP from 90.9% to 83.5%). The most stringent 25% gate again lowers performance, decreasing a further 10-12 points for Gemini-2.0

Flash SP from 94.3% to 73.9%, and GPT-4o SC from 87.2% to 65.7%. This progressive decline reflects the accuracy of each model decreasing as questions get harder and harder to generate.

The open-source set also demonstrates an equally robust sensitivity to threshold decrease. AIN begins well at 78.5% SC with no gate, drops to 69.1% at 50%, and further to 58.2% at 25%, a 20-point decline. Aya Vision's decline is equally steep, dropping from 77.0% to 68.7% (50%) and further to 57.3% (25%). Qwen 2 VL and Qwen 2.5 VL, although some of the strongest open models, follow this trend too: Qwen 2.5 VL's SP accuracy goes from 91.5% (No Gate) to 82.0% (50%) and then from 71.5% (25%). Even top models lose around a 20-point difference under the toughest threshold. This cascading decline across open-source architectures reveals that with increasingly harder tasks, model confidence is lower.

8 Limitations

Although our self-learning Arabic QA system provides strong empirical gains and automaton benefits, its shortcomings remain. To begin with, as compelling as the system's performance on long-context Arabic documents is, its quality is highly sensitive to the structure and quality of input documents. High-visual-noise, scan-degraded, or non-standard layout documents, a common feature of historical Arabic collections, are capable of compromising the fidelity of the layout parser and impacting downstream QA accuracy.

Second, while fully automated workflow, current LVLM-based system relies on strict prompting templates and hard-coded complexity bounds as thresholds for validity checking and tuning. Future

updates would involve reinforcement learning or adaptive policy selection mechanisms in an effort to make more intelligent and adaptive prompting strategies.

Third, computational cost and system complexity are not to be underestimated. The multi-LVLM iterative pipeline, particularly in the self-refining stages, may be quite expensive in terms of latency and hardware. This can pose difficulties for real-time or mass deployment, especially where resources are limited.

Lastly, while we designed our architecture with Arabic-specific challenges in mind (e.g., bidirectional text, script variability), it remains to be seen how the system will perform across dialectal forms, handwriting material, or low-resource scripts generally within the broader Arabic linguistic context. Accommodating diverse regional Arabic dialects and mixed-script material remains an important area for further research.

9 Conclusion

In this work, we introduced a self-evolving adversarial pipeline. Through the integration of state-of-the-art structural layout analysis and preprocessing, our pipeline not only successfully scales across diverse long-form texts but also strictly follows source content with fidelity in each iterative cycle.

One of the highlights of our system is its adaptive level of difficulty, enabling data generation from trivial recall drills to very advanced inference problems. The capability is present naturally for enabling curriculum-learning approaches, incrementing task difficulty over time in an effort to maximize model calibration and learning efficiency.

The methodology will be applied to other domains and modalities in the future, be computationally optimized, and be introduced with adaptive thresholding for adaptive data generation.

In addition, we have developed an automated data-acquisition pipeline that transforms a single, high-level user query into a fully automated, multi-stage harvesting process capable of gathering millions of documents in a matter of hours with an easy-to-use interface and end-to-end automation that eliminates manual data-collection bottlenecks. Because of its agent-based, modular design, the pipeline is readily extensible to new domains and languages far beyond the scope of Arabic document understanding such as legal rulings, medical literature reviews, or multilingual scientific benchmarks.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2024. Claude 3 haiku: Our fastest model yet. Accessed: 2025-02-11.

Jeremy Belval. 2018. pdf2image: A python library to convert pdf pages to images using poppler. GitHub repository.

Gagan Bhatia, El Moatez Billah Nagoudi, Fakhraddin Alwajih, and Muhammad Abdul-Mageed. 2024. Qalam: A multimodal llm for arabic optical character and handwriting recognition. *arXiv preprint arXiv:2407.13559*.

Yew Ken Chia, Liying Cheng, Hou Pong Chan, Chaoqun Liu, Maojia Song, Sharifah Mahani Aljunied, Soujanya Poria, and Lidong Bing. 2024. M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. *arXiv preprint arXiv:2411.06176*.

Chao Deng, Jiale Yuan, Pi Bu, Peijie Wang, Zhong-Zhi Li, Jian Xu, Xiao-Hui Li, Yuan Gao, Jun Song, Bo Zheng, and 1 others. 2024. Longdocurl: a comprehensive multimodal long document benchmark integrating understanding, reasoning, and locating. *arXiv preprint arXiv:2412.18424*.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, and 1 others. 2024. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *arXiv preprint arXiv:2404.06512*.

Gemini Team. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. Accessed: 2025-02-11.

Sara Ghaboura, Ahmed Heakl, Omkar Thawakar, Ali Alharthi, Ines Riahi, Abduljalil Saif, Jorma Laaksonen, Fahad S Khan, Salman Khan, and Rao M Anwer. 2024. Camel-bench: A comprehensive arabic lmm benchmark. *arXiv preprint arXiv:2410.18976*.

Ahmed Heakl, Abdullah Sohail, Mukul Ranjan, Rania Hossam, Ghazi Ahmed, Mohamed El-Geish, Omar Maher, Zhiqiang Shen, Fahad Khan, and Salman Khan. 2025. Kitab-bench: A comprehensive multi-domain benchmark for arabic ocr and document understanding. *arXiv preprint arXiv:2502.14949*.

Benjamin Kiessling, Daniel Stökl Ben Ezra, and Matthew Thomas Miller. 2019. Badam: a public dataset for baseline detection in arabic-script manuscripts. In *Proceedings of the 5th International Workshop on Historical Document Imaging and Processing*, pages 13–18.

Labellerr Inc. 2024. Labellerr: Automated activelearning and selfcorrection annotation pipeline. Proprietary product documentation. Available from Labellerr Inc.

LandingAI. 2025. Agentic document extraction. Proprietary product documentation. Available from LandingAI.

Bo Li, Kaichen Zhang, Hao Zhang, Dong Guo, Renrui Zhang, Feng Li, Yuanhan Zhang, Ziwei Liu, and Chunyuan Li. 2024. Llava-next: Stronger llms supercharge multimodal capabilities in the wild.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, and 1 others. 2024. Mmlongbenchdoc: Benchmarking long-context document understanding with visualizations. *arXiv preprint arXiv:2407.01523*.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 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.

Omer Nacar, Yasser Al-Habashi, Serry Sibaee, Adel Ammar, and Wadii Boulila. 2025. Sard: A large-scale synthetic arabic ocr dataset for book-style text recognition. *arXiv preprint arXiv:2505.24600*.

OpenAI. 2024a. Advancing reasoning with o3. Accessed: 2025-02-12.

OpenAI. 2024b. Introducing o1: A new era of reasoning models. Accessed: 2025-02-12.

Mohamed Rashad. 2024. Arabic-nougat: Fine-tuning vision transformers for arabic ocr and markdown extraction. *arXiv preprint arXiv:2411.17835*.

Leonard Richardson. 2007. Beautiful soup documentation.

Rana SM Saad, Randa I Elanwar, NS Abdel Kader, Samia Mashali, and Margrit Betke. 2016. Bce-arabic-v1 dataset: Towards interpreting arabic document images for people with visual impairments. In *Proceedings of the 9th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, pages 1–8.

Zach Smiley. 2020. pdfplumber: A Python library for extracting text and tables from pdfs. https://github.com/jsvine/pdfplumber.

Ray Smith. 2007. An overview of the tesseract ocr engine. In *Proceedings of the International Conference on Document Analysis and Recognition*, pages 629–633.

Tarjama (Arabic.AI). 2025. Arabic.ai agentic ecosystem. Proprietary product documentation. Available from Arabic.AI.

UiPath. 2025. Activelearning document understanding pipeline. Proprietary product documentation. Available from UiPath.

Unicode Consortium. 1996. *The Unicode Standard, Version 2.0*. AddisonWesley.

Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, and 1 others. 2023. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, Chunrui Han, and Xiangyu Zhang. 2024a. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *Preprint*, arXiv:2409.01704.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, and 1 others. 2024b. General ocr theory: Towards ocr-2.0 via a unified end-to-end model.

Wenda Xu, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, Biao Zhang, Zhongtao Liu, William Yang Wang, Lei Li, and Markus Freitag. 2023. Fine-grained llm agent: Pinpointing and refining large language models via fine-grained actionable feedback. *arXiv preprint arXiv:2311.09336*.

Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. 2024. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *arXiv preprint arXiv:2410.12628*.

Fengbin Zhu, Wenqiang Lei, Fuli Feng, Chao Wang, Haozhou Zhang, and Tat-Seng Chua. 2022. Towards complex document understanding by discrete reasoning. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4857–4866.