# A Data Annotation Approach Using Large Language Models

Carlos Rocha[a], Jonatas Grosman[b], Fernando A. Correia[c], Venicius Rego[d] and Hélio Lopes[e]
*Department of Informatics, PUC-Rio, Marquês de São Vicente, 225 RDC, 4th floor - Gávea, Rio de Janeiro, Brazil*

Keywords: Data Annotation, Large Language Model, Visual Question-Answering, Documents, Machine Learning.

Abstract: Documents are crucial for the economic and academic systems, yet extracting information from them can be complex and time-consuming. Visual Question Answering (VQA) models address this challenge using natural language prompts to extract information. However, their development depends on annotated datasets, which are costly to produce. To face this challenge, we propose a four-step process that combines Computer Vision Models and Large Language Models (LLMs) for VQA data annotation in financial reports. This method starts with Document Layout Analysis and Table Structure Extraction to identify document structures. Then, it uses two distinct LLMs for the generation and evaluation of question and answer pairs, automating the construction and selection of the best pairs for the final dataset. As a result, we found Mixtral-8x22B and GPT-4o mini to be the most cost-benefit for generating pairs, while Claude 3.5 Sonnet performed best for evaluation, aligning closely with human assessments.

## 1 INTRODUCTION

Data annotation is essential for developing supervised machine learning models, especially in Natural Language Processing (NLP) and Computer Vision (CV). The quality of these models relies on large volumes of labeled data for tasks like document comprehension and interpretation. Document Visual Question Answering (DocVQA) exemplifies this by combining NLP and CV to enable models to interpret document images, leveraging textual and visual features to allow interaction with document information through natural language queries.

Training DocVQA models requires thousands of carefully reviewed and annotated document pages (Mathew et al., 2021). This annotation and review process is time-consuming and costly, primarily due to the significant human resources needed. To meet this demand, the literature often employs non-expert annotators through crowd-sourcing platforms like Amazon Mechanical Turk (Mathew et al., 2021; Mathew et al., 2022). However, using such platforms presents significant challenges and limitations, including inconsistent annotation quality and the additional effort required for review and validation, increasing both cost and time (Kittur et al., 2008).

DocVQA models are used in domains like scientific research, insurance, and finance for tasks such as entity relationship identification, policy comprehension, fraud detection, and financial data extraction. Financial reports, such as those from the Brazilian Stock Market (*Brasil Bolsa Balcão - B3*)[1], are a valuable data source. In B3, 2,706 companies are mandated by the Brazilian Securities and Exchange Commission, or *Comissão de Valores Mobiliários* (CVM),[2] to publish annual balance sheets, quarterly reports, and other information, reflecting similar frameworks as the Securities and Exchange Commission (SEC)[3] in the U.S. and European Securities and Markets Authority (ESMA)[4] in the EU.

Although these financial reports are public, they lack structured annotated data, requiring solutions to the data annotation challenge. One approach is using Large Language Models (LLMs) as annotators to generate labels in a zero-shot or few-shot manner. While promising, this method introduces noise, particularly in complex, domain-specific tasks (Agrawal et al., 2022). Given their remarkable capability in text annotation, LLMs can automate the creation of anno-

[a] https://orcid.org/0009-0004-9696-330X
[b] https://orcid.org/0000-0002-1152-5828
[c] https://orcid.org/0000-0003-0394-056X
[d] https://orcid.org/0000-0002-7101-4608
[e] https://orcid.org/0000-0003-4584-1455

[1] https://www.b3.com.br/
[2] https://www.gov.br/cvm/en
[3] https://www.sec.gov/
[4] https://www.esma.europa.eu/



Rocha, C., Grosman, J., Correia, F. A., Rego, V. and Lopes, H.
A Data Annotation Approach Using Large Language Models.
DOI: 10.5220/0013280100003929
In *Proceedings of the 27th International Conference on Enterprise Information Systems (ICEIS 2025) - Volume 1*, pages 748-755
ISBN: 978-989-758-749-8; ISSN: 2184-4992

tated question-answer datasets for DocVQA. Leveraging this generalist capability can significantly reduce time and cost, leading to our main research question: *MRQ: How can we use Large Language Models in data annotation for Document Visual Question Answering task?*.

To address *MRQ*, the approach must handle diverse document layouts, ensuring readable text representation and reliable, high-quality question-answer pairs for effective model training. This leads to two sub-questions: *RQ1: How can we combine computer vision models and Large Language Models to generate questions and answers from documents?*, and *RQ2: How can we assess the quality of the questions and answers generated?*.

We propose a three-stage process: (i) Transcription, (ii) Question-Answer Generation, and (iii) Question-Answer Judgment. First, we transcribe the document page by recognizing the characters and the layout's main points and converting the tables into a markup language. The transcriptions are then inputted into an LLM to generate QA pairs, followed by a final stage where another LLM evaluates the quality of these pairs for inclusion in the final dataset.

Our work's main contributions are the following:

- We present a new approach for generating and evaluating question-answer pairs in documents that contain both text and tables.
- We contribute by evaluating different models for Document Layout Analysis and Table Structure Recognition to extract information from B3's financial reports.

The remainder of this paper is organized as follows: Section 2 reviews key works on DocVQA dataset construction and LLM data annotation. Section 3 outlines our method and its stages. Section 4 details the experiments validating the models and approach. Finally, Section 5 highlights findings, limitations, and future work.

## 2 RELATED WORK

DocVQA requires a labeled dataset for effective model training as in any supervised machine learning task. High-quality data helps models interpret various textual, structural, and graphical elements in a document. (Mathew et al., 2021) introduced a dataset with complex documents containing 50,000 questions over 12,767 images, curated from the UCSF Industry Document Library and annotated through a three-stage crowdsourced process of question generation, validation, and review. Our approach is similar to (Mathew et al., 2021) but with fewer questions per page, adjusted validation, and using an LLM to judge the QA pairs generated.

LLMs have been explored for efficient annotation in NLP. (Wang et al., 2021) showed GPT-3 significantly reduced costs in question generation while maintaining human-like quality. Our work aligns with theirs, focusing on financial reports in QA annotation with LLMs. For complex table data, (Nguyen et al., 2023) converted tables to HTML and dataframes for QA input. Given LLMs tendency to hallucinate, question validity is crucial. (Bai et al., 2024) introduced a benchmarking framework where LLMs both generate and evaluate questions, reducing biases via decentralized peer evaluation. Larger models like LLaMA-65B and GPT-4 showed enhanced accuracy in this setup. We adopt this evaluation framework using LLMs to assess annotations, filtering errors and improving automation without human intervention.

## 3 METHODOLOGY

As depicted in Figure 1, the process consists of three stages: (i) Transcription, (ii) Question-Answer Generation, and (iii) Question-Answer Judgment. The transcription stage generates a textual representation of document files and feeds an LLM to generate QA tuples in the second stage. Finally, the same transcription and the generated QA tuples feed another LLM to select the valid tuples to compose the dataset.

### 3.1 Transcription

The process begins with the Transcription Stage, where a document file is received and the desired pages are extracted as images for subsequent steps. An OCR tool is then applied to transcribe the textual content of these images.

Given the importance of OCR and the lack of annotated data for training and evaluating open-source tools, we selected Microsoft OCR V4.[5] This choice was based on the work of (Santos et al., 2023), whose evaluation of various OCR tools on diverse Portuguese text images showed that the Microsoft model outperformed all other evaluated models, delivering solid results even with different backgrounds and text rotation levels.

The Document Layout Analysis (DLA) step uses an object detection model for structure recognition. Object detection models can generally identify which

[5]https://azure.microsoft.com/en-us/products/ai-services/ai-document-intelligence

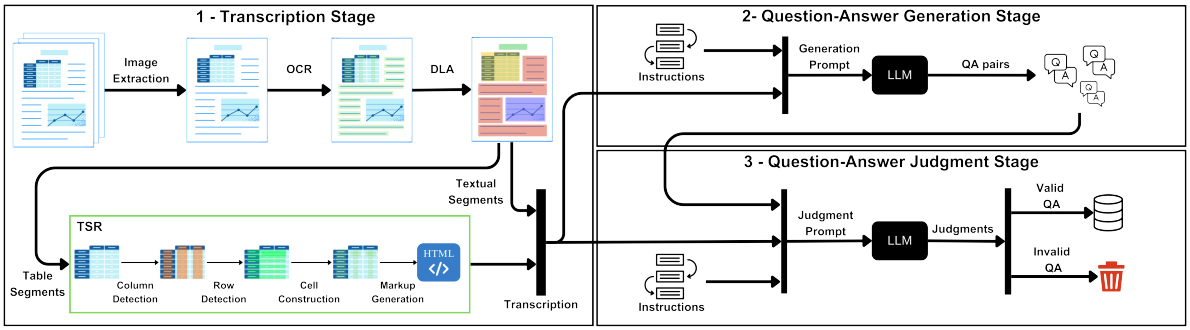


Figure 1: Proposed process overview.

objects from a known set are present in the image and provide information about their position. In our case, the objects are document segments like tables, images, and section headers, with the DLA model returning the bounding boxes of the segments.

After segmentation, the identified regions were stored in a data structure containing their bounding boxes, the detected class, positional information, and the textual content detected by the OCR tool. The intersection of the text and region bounding boxes was calculated to determine if the text was within a specific region. Given the variety of available DLA models, we selected models from the literature and performed experiments to choose the best one, which we describe in more detail in Subsection 4.1.

The Table Structure Recognition (TSR) step uses a second object detection model to segment table structures, detecting rows and columns as bounding boxes and defining cells based on the intersections. Next, a separate algorithm converts these bounding boxes into HTML. As with the DLA step, multiple models are available. So, we reviewed models from the literature and conducted experiments to select the best one, detailed in Subsection 4.2.

At the end of the Transcription stage, the document image transcription is modified to include positional markers. These markers identify the transcript regions used to generate a QA pair. For text regions, we add the paragraph number; for table regions, we add the table number and the number of each row.

## 3.2 Question-Answer Generation

The modified document transcription from the previous step is input into the selected LLM for the Question-Answer Generation stage, along with generation instructions. [6] These instructions include the coherence standards the model should follow, rules for using relative pronouns, the number of desired tuples, the focus area, and the key elements to include in the output. The prompt also provides positive and negative examples to guide the generation process. Finally, the expected output of this stage is a textual format divided into three parts: QUESTION - ANSWER - TEXT REGION. The first two form the QA tuple, while the third is a positional marker indicating the text region used for generation.

## 3.3 Question-Answer Judgment

Finally, the objective of the Question-Answer Judgment stage is to ensure that the generated questions and answers are coherent and correct, filtering out possible hallucinations from the generating model and increasing the quality of the final dataset. This stage also receives as input the transcript of the document, without the positional markers, together with the evaluation instructions, [7] specifying the criteria for evaluating the questions and answers.

Each QA tuple judgment uses two calls to the LLM. The first call evaluates the coherence of the question, which is considered coherent if the questions comply with the correct use of the grammatical rules of the Portuguese language and have only one answer. The second call evaluates the answer to see if it answers the question. Since each QA tuple is evaluated individually, the process described is carried out in 3 iterations per document page. The expected output for each iteration is binary, confirming or denying the coherence of the question and the validity of the provided answer.

[6] Generation instructions available on Github: *https://github.com/carlos-vinicios/DocVQA-Data-Annotator*

[7] Evaluation instructions available on Github: *https://github.com/carlos-vinicios/DocVQA-Data-Annotator*

# 4 EXPERIMENTS

This section describes the experiments conducted following the proposed process. We begin with the experimentation of the transcription stage, where we evaluated available DLA models from the literature using a dataset created during this study. Next, we assessed open-source TSR models, testing them on four datasets to measure their generalization capabilities. After the transcription stage, we proceeded to evaluate various LLMs for the QA generation stage. Finally, we assessed the most robust LLMs for the QA tuple judgment stage.

## 4.1 Document Layout Analysis (DLA)

To evaluate the DLA models, we built an annotated dataset using a process similar to the creation of the DocLayNet dataset (Pfitzmann et al., 2022). The validation dataset consisted of 990 randomly selected pages from financial statements, focusing on central pages that typically contain text, tables, and images.

Two annotators independently annotated each page, following DocLayNet's criteria, and reviewed each other's work to improve consistency. The Model evaluation was based on the Mean Average Precision (mAP), a standard metric for object detection and segmentation. The average Precision (AP) was calculated for each class as the area under the precision-recall curve. The model selection followed two main criteria: it needed to be trained on the DocLayNet dataset and be open-source, providing access to its weights for evaluation and modification.

Table 1 shows the mAP performance for each class. RoDLA achieved the highest overall performance, excelling in handling diverse document types and perturbations. Malaysia-AI performed well in most classes, especially in detecting tables and text, benefiting from advanced data augmentation techniques despite using the same architecture as Maik Thiele. SwinDocSegmenter, although computationally demanding, showed lower performance than Malaysia-AI, which can run efficiently on a CPU.

In conclusion, RoDLA is the best choice for the DLA task. While it has a longer execution time, its superior ability to identify diverse document regions ensures more accurate layout segmentation,

## 4.2 Table Structure Recognition (TSR)

Building a specific dataset for TSR model evaluation is unfeasible due to the high time cost of data annotation and refinement, wich involves marking column and row bounding boxes, transcribing table text, and creating the HTML structure. Three datasets were selected for TSR evaluation to address this. The first is PubTabNet, with around 500,000 table images and their HTML equivalents for the structural recognition of tables (Zhong et al., 2020). FinTabNet, which focuses on financial documents, provides annotated data for table extraction and conversion to HTML (Zheng et al., 2021). Finally, the ICDAR 2013 Table Competition dataset serves as a benchmark for this task(Göbel et al., 2013).

We used TEDS-Struct, a variant of Tree-Edit-Distance-based Similarity (TEDS) (Zhong et al., 2020), to evaluate table extraction model performance. Unlike the original TEDS, which assesses structure and text alignment, TEDS-Struct focuses solely on the HTML structure, ignoring transcription errors (Qiao et al., 2021), making it suitable for this work. For TSR model selection, we identified open-source models capable of table-to-HTML conversion from table images. Four models were chosen: Table Transformer (TATR) (Smock et al., 2022); MLT-TabNet (Ly and Takasu, 2023); Table Master (Ye et al., 2021), and Local and Global Pyramid Mask Alignment (LGPMA) (Qiao et al., 2021).

Table 2 shows the results for each tested model. Although no model excelled across all datasets, TATR All and MLT-TabNet demonstrated the best generalization abilities, with TATR All achieving the best performance on the ICDAR 2013 benchmark dataset. The findings suggest models trained on the FinTabNet dataset exhibit superior generalization, likely due to the dataset's complexity and variety of examples featuring different degrees of customization. In contrast, PubTabNet is limited to tables from scientific publications, offering less diverse training data.

Given its extraction performance, TATR All is the best model for this work. It showed strong generalization abilities, performing well across all tested datasets, was the fastest model, and the only one capable of maintaining a low execution time on CPU.

## 4.3 Question-Answer Generation

We selected models based on cost-benefit, considering the token usage required to complete the task. As this stage generates the most output tokens, it is the main cost driver. On average, document transcriptions and task instructions use $3,471(\pm 1080)$ input tokens, while the question, answer, and positional marker produce $170(\pm 36)$ output tokens.

Figure 2 shows the quality and cost (USD per million tokens) of the main available models, assuming a 3:1 ratio of input to output tokens. The price format and quality were based on (Analysis, 2024). In the

Table 1: mAP performance for each class and model on a constructed dataset. The best performance is formatted in boldface.

| Class | RoDLA (Chen et al., 2024) | Malaysia-AI (Malaysia-AI, 2024) | SwinDocSegmenter (Banerjee et al., 2023) | Maik Thiele (Maik Thiele, 2024) |
|---|---|---|---|---|
| Caption | **0.302** | 0.015 | 0.067 | 0.127 |
| List Item | **0.860** | 0.741 | 0.676 | 0.642 |
| Picture | **0.749** | 0.413 | 0.569 | 0.524 |
| Text | **0.979** | 0.910 | 0.776 | 0.901 |
| Footnote | **0.735** | 0.486 | 0.345 | 0.254 |
| Page Footer | **0.911** | 0.446 | 0.621 | 0.435 |
| Section Header | **0.776** | 0.490 | 0.391 | 0.346 |
| Title | 0.007 | **0.015** | 0.003 | 0.003 |
| Formula | - | - | - | - |
| Page Header | **0.870** | 0.476 | 0.485 | 0.447 |
| Table | **0.975** | 0.955 | 0.899 | 0.906 |
| All | **0.716** | 0.495 | 0.483 | 0.459 |

Table 2: TEDS-Struct performance by model on each dataset. The best performance is formatted in boldface.

| Model | FinTabNet | PubTabNet | ICDAR 2013 |
|---|---|---|---|
| TATR All | 0.92 | 0.93 | **0.94** |
| TATR Fin | 0.92 | 0.88 | 0.91 |
| MLT-TabNet | **0.97** | 0.90 | 0.91 |
| TableMaster | 0.80 | **0.97** | 0.90 |
| LGPMA | 0.41 | 0.96 | 0.91 |

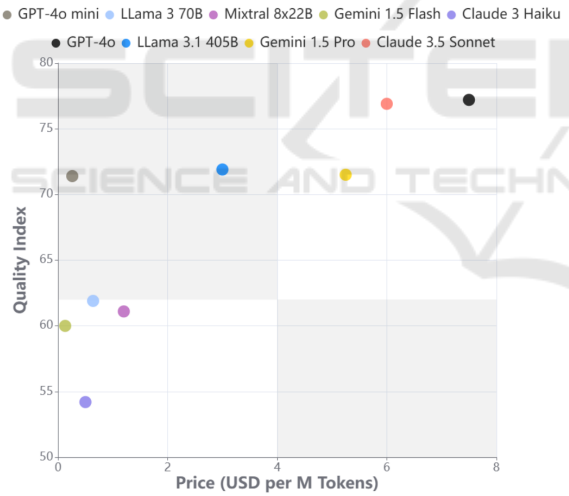


Figure 2: Quality vs Price for the main LLM models available via a pay-per-use API. The price of each model was collected in August 2024.

Figure, models are grouped into four quadrants: high quality and high cost, high quality and low cost, reasonable quality and low cost, and lower-performing models with varying costs.

Based on the average token requirements and the price-quality balance in Figure 2, we selected models from the second and third quadrants: GPT-4o mini, LLaMA 3 70B, Mixtral 8x22B, Gemini 1.5 Flash, and Claude 3 Haiku. LLaMA 3.1 405B, the best but most expensive open-source model, was excluded from this stage and reserved for the third stage of our process.

For the question generation experiments using LLM, we created a dataset of 300 randomly selected non-consecutive pages, which feature only text, only tables or a mixture of both. The tables have different structures and sizes, so the complexity of interpreting each one varies, with larger tables being more complex. The information on each page may neither start nor end within that page.

The first experiment assessed the distribution of questions based on their initial 3-grams. The goal was to evaluate the models' ability to generate diverse questions. Figure 3 shows the distribution of the generated questions, where each level represents the most frequent words in the natural reading order. Thus, the first level corresponds to the first word of the question, the second level to the second word, and so on. For example, Mixtral 8x22B frequently starts questions with "Qual foi" ("What was") or "Qual é o" ("What is the"), while empty regions at the second level indicate infrequent words.

Most models used at least five different interrogative pronouns, but Cluade 3 Haiku and Gemini 1.5 Flash showed less variety, favoring a more conservative question phrasing approach. Also, most models favored using *"Qual"* ("What") as the interrogative pronoun, likely due to its adaptability in various contexts. This pattern aligns with human-generated questions, as demonstrated by (Mathew et al., 2021). *"Quanto"* ("how much" or "how many") was the second most frequently used pronoun, linked to the prevalence of tables in the selected pages. Lastly, *"Quando"* ("when") and *"Quem"* ("Who"), though not explored equally by all models, were employed by GPT-4o mini and Mixtral-8x22B.

Based on the variation, GPT-4o mini and Mixtral-8x22B showed the most promise for question generation. Considering the cost-performance trade-off, Mixtral-8x22B is the optimal choice, being open-

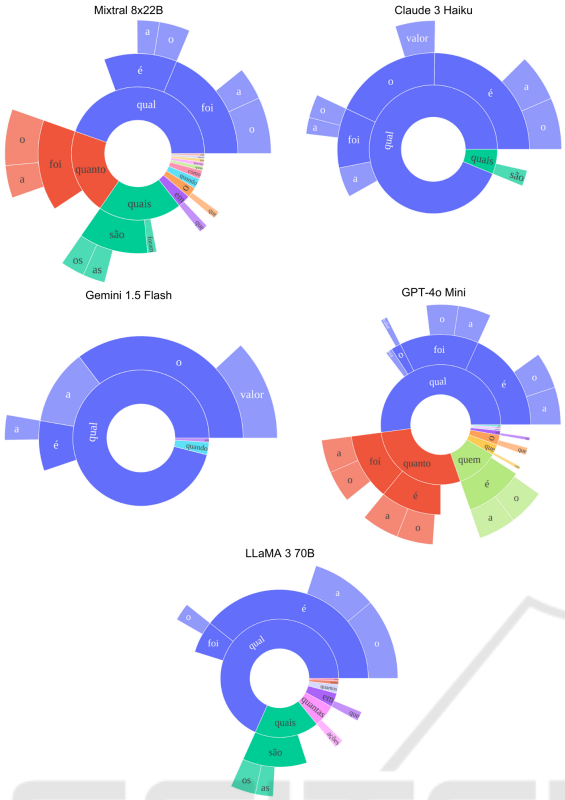


Figure 3: Distribution of questions by their starting 3-grams.

source and offering strong performance at a lower cost. To evaluate the quality of the questions and answers generated by the models, we conducted a human evaluation with 28 voluntary annotators. They assessed each QA tuple using a web tool developed for this research. Annotators evaluated the coherence of the questions, ensuring they were understandable, adhered to Portuguese language norms, and were free of ambiguity. If the question was coherent, they validated whether the answers were correct based on the document information. The validation process was binary: annotators responded ”yes” or ”no” for both criteria. A question was considered valid if it was coherent and the answer was correct.

The evaluation was conducted in batches of 10 pages, with each page containing three QA tuples. Annotators reviewed each page individually. Some annotators were responsible for more than one batch due to the number of annotators and available files. To maintain consistency, each batch was reviewed by at least three distinct annotators and inter-annotator agreement was measured.

The average percent inter-annotator agreement is 0.84 ($\pm$0.29) for question coherence and 0.84 ($\pm$0.28) for response accuracy. Based on this high level of agreement and the number of annotators, we retained all annotations and used them to validate the models.

Table 3: Question coherence, answer accuracy, and QA tuples valid proportion for each LLM.

| Model | Question Coherence | Answers Accuracy | QA Tuples Valid |
|---|---|---|---|
| LLaMA 3 70B | 0.89 | **0.97** | **0.87** |
| Mixtral 8x22B | 0.89 | 0.95 | 0.85 |
| Gemini 1.5 Flash | **0.93** | 0.91 | 0.84 |
| GPT-4o mini | 0.83 | **0.97** | 0.81 |
| Claude 3 Haiku | 0.85 | 0.94 | 0.79 |

Using this dataset, we evaluated the model performance based on the proportion of coherent questions and correct answers in the QA tuples. Table 3 shows that LLaMA 3 70B generated 0.89 coherent questions, of which 0.97 had correct answers, yielding 0.87 valid tuples. For all models, the final dataset included only tuples with coherent questions and accurate answers.

The results showed that LLaMA 3 70B excelled in coherence and accuracy, while Mixtral-8x22B maintained strong performance with higher QA tuple generation. Gemini 1.5 Flash was less consistent, particularly in answer accuracy, and Claude 3 Haiku displayed the highest variation, indicating randomness in its generation process. GPT-4o mini was stable with increased tuple output. Overall, LLaMA 3 70B and Mixtral-8x22B were the most effective models.

## 4.4 Question-Answer Judgment

For the QA judgment stage, we prioritized high-quality models, as this step is crucial for filtering out invalid questions automatically. To ensure accurate assessment of the generated QA tuples, we needed models with strong general task performance and the ability to understand document transcriptions to ensure accurate evaluations and reliable use of the generated tuples.

At this stage, cost primarily stems from input tokens, as the instruction prompt is resent for each QA tuple evaluation. The average input size was $2261.33 \pm 819.75$ tokens, while the output averaged $2.67 \pm 1.72$ tokens. We selected three high-quality models from the first quadrant: GPT-4o, Claude 3.5 Sonnet, and Gemini 1.5 Pro. Additionally, we included LLaMA 3.1 405B, a leading open-source model, to compare commercial and open-source options.

To evaluate this stage, we used the dataset created after a human evaluation during the QA generation stage, described in Subsection 4.3. We used this dataset to assess the judgment capabilities of the pro-

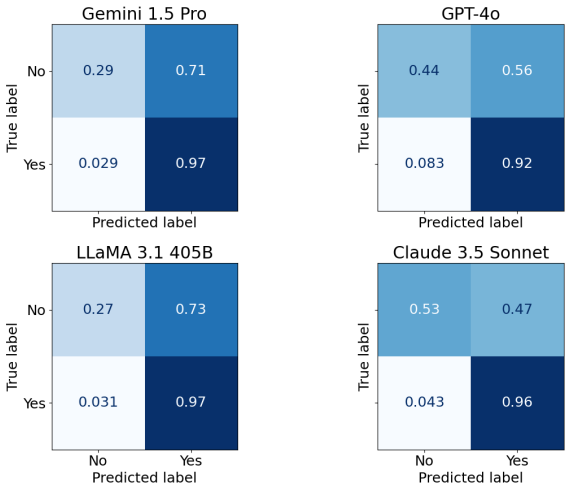


Figure 4: Confusion matrix between LLM and human for the validity of a QA tuple. The human annotation is considered the true label.

posed approach by comparing the performance of an LLM and a human annotator. As in the second stage (Question-Answer Generation), we used Langchain to normalize the API calls, setting the temperature to zero and all the other parameters to default. The LLM judgment of QA tuples followed the same criteria as the human evaluation, assessing question coherence, adherence to Portuguese language norms, and clarity. Models also validated answers accuracy based on the document transcriptions. The judgment process was binary, with the LLM responding "yes" or "no". A question was deemed valid if it was coherent and the answer accurate.

After testing all models, we compared their F1 scores and alignment with human judgments for final dataset use. All four models performed well, with F1 scores above 0.8. The best performance came from Claude 3.5 Sonnet, which achieved the highest score at 0.93, followed by Gemini 1.5 Pro and LLaMA 3.1 405B, both at 0.92, while GPT-4o scored 0.90.

The confusion matrix in Figure 4 reveals that all models struggled to identify false sentences, likely due to the higher proportion of negative samples in the dataset, which made detecting invalid tuples more challenging. To improve judgment performance, we combined the models into an ensemble. In this approach, a QA tuple was considered valid only if all models confirmed both the coherence of the question and the accuracy of the answer. Table 4 shows the F1 obtained by each ensemble, where we can see that the combination of models did not improve the metric, showing a similar performance for all combinations.

Table 4: F1 score between an ensemble of LLMs and humans for the validity of a QA tuple.

| Ensemble | Gemini 1.5 Pro | GPT4-o | LLaMA 3.1 405B | Claude 3.5 Sonnet | F1 |
|---|---|---|---|---|---|
| 1 | ✓ | ✓ | ✓ | | 0.82 |
| 2 | ✓ | ✓ | | ✓ | 0.82 |
| 3 | ✓ | | ✓ | ✓ | 0.83 |
| 4 | | ✓ | ✓ | ✓ | 0.82 |

# 5 CONCLUSION

This paper addresses the research questions by developing a data annotation process for the DocVQA task using LLMs. To tackle the first part of *RQ1*, we designed a three-step process employing CV models to extract textual representations from documents. For the second part, we evaluated five cost-effective LLMs for generating QA pairs based on document transcriptions and predefined rules to ensure well-formed questions and enable human evaluation. To answer *RQ2*, we conducted a human evaluation to assess the validity of QA pairs generated by each LLM, measuring the validity-to-invalidity ratio. Additionally, we evaluated four robust LLMs for automating the validation process, benchmarking their performance against human assessments.

Our findings demonstrate that LLMs can effectively generate QA pairs with a favorable validity-to-invalidity ratio. Mixtral-8x22B and GPT-4o mini achieved the best results, generating diverse, instruction-compliant QA pairs with minimal invalid outputs, with GPT-4o mini excelling in cost-effectiveness. For automatic QA pair validation, Claude 3.5 Sonnet achieved the highest F1 score (0.93) but requires further improvements to reduce false negatives. Ensemble models were tested as an enhancement but underperformed compared to individual models, achieving a maximum F1 score of 0.83 with significantly higher costs, making them impractical despite a slight reduction in false positives.

## 5.1 Future Works and Limitations

As discussed earlier, the process occasionally fails in the validity judgment of QA pairs. To improve the validity judgment of QA pairs, we plan to introduce a review score for each generated pair in future works. This score will help identify questions requiring human review, minimizing false positives and negatives while enhancing dataset robustness. Additionally, we intend to conduct a qualitative assessment of question relevance and structure, expanding our focus beyond

validity analysis to ensure alignment with the target domain.

Given the process's reliance on document transcription, we also aim to investigate the impact of transcription errors on question generation. This analysis will provide insights into model performance under such conditions and inform strategies to mitigate these effects. Ultimately, we aim to produce a fully annotated dataset using the proposed process, establish baselines with state-of-the-art DocVQA models, and evaluate the process's strengths and limitations for further refinement.

## ACKNOWLEDGMENTS

This study was supported by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

## REFERENCES

Agrawal, M., Hegselmann, S., Lang, H., Kim, Y., and Sontag, D. (2022). Large language models are few-shot clinical information extractors. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Analysis, A. (2024). Independent analysis of ai models and api providers. https://artificialanalysis.ai/. Accessed: 2024-08-20.

Bai, Y., Ying, J., Cao, Y., Lv, X., He, Y., Wang, X., Yu, J., Zeng, K., Xiao, Y., Lyu, H., et al. (2024). Benchmarking foundation models with language-model-as-an-examiner. *Advances in Neural Information Processing Systems*, 36.

Banerjee, A., Biswas, S., Lladós, J., and Pal, U. (2023). Swindocsegmenter: An end-to-end unified domain adaptive transformer for document instance segmentation. In *International Conference on Document Analysis and Recognition*, pages 307–325. Springer.

Chen, Y., Zhang, J., Peng, K., Zheng, J., Liu, R., Torr, P., and Stiefelhagen, R. (2024). Rodla: Benchmarking the robustness of document layout analysis models. *arXiv preprint arXiv:2403.14442*.

Göbel, M., Hassan, T., Oro, E., and Orsi, G. (2013). Icdar 2013 table competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1449–1453. IEEE.

Kittur, A., Chi, E. H., and Suh, B. (2008). Crowdsourcing user studies with mechanical turk. In *Proceedings of the SIGCHI conference on human factors in computing systems*, pages 453–456.

Ly, N. T. and Takasu, A. (2023). An end-to-end multi-task learning model for image-based table recognition. pages 626–634.

Maik Thiele (2024). documentlayoutsegmentation_yolov8_ondoclaynet (revision 25486d5).

Malaysia-AI (2024). Yolov8x-doclaynet-full-1024-42.

Mathew, M., Bagal, V., Tito, R., Karatzas, D., Valveny, E., and Jawahar, C. (2022). Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

Mathew, M., Kondreddi, V. K., Biten, A. F., Mafla, A., Matas, J., Jawahar, C. V., Valveny, E., and Karatzas, D. (2021). Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2200–2209.

Nguyen, P., Ly, N. T., Takeda, H., and Takasu, A. (2023). Tabiqa: Table questions answering on business document images. *arXiv preprint arXiv:2303.14935*.

Pfitzmann, B., Auer, C., Dolfi, M., Nassar, A. S., and Staar, P. (2022). Doclaynet: a large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3743–3751.

Qiao, L., Li, Z., Cheng, Z., Zhang, P., Pu, S., Niu, Y., Ren, W., Tan, W., and Wu, F. (2021). Lgpma: complicated table structure recognition with local and global pyramid mask alignment. In *International conference on document analysis and recognition*, pages 99–114. Springer.

Santos, Y., Silva, M., and Reis, J. C. S. (2023). Evaluation of optical character recognition (ocr) systems dealing with misinformation in portuguese. In *2023 36th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 223–228.

Smock, B., Pesala, R., and Abraham, R. (2022). Pubtables-1m: Towards comprehensive table extraction from unstructured documents. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4634–4642.

Wang, S., Liu, Y., Xu, Y., Zhu, C., and Zeng, M. (2021). Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Ye, J., Qi, X., He, Y., Chen, Y., Gu, D., Gao, P., and Xiao, R. (2021). Pingan-vcgroup's solution for icdar 2021 competition on scientific literature parsing task b: table recognition to html. *arXiv preprint arXiv:2105.01848*.

Zheng, X., Burdick, D., Popa, L., Zhong, X., and Wang, N. X. R. (2021). Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 697–706.

Zhong, X., ShafieiBavani, E., and Jimeno Yepes, A. (2020). Image-based table recognition: data, model, and evaluation. In *European conference on computer vision*, pages 564–580. Springer.