Model Performance for PDF-to-Text Processing

Pablo de Vicente Abad 8/04/2025

Overview

This report evaluates the performance of various models used in our PDF-to-text conversion workflow. The workflow is composed of three main components:

- **Raw Text Information**
- **Table-to-Text Conversion**
- **Image-to-Text Extraction**

Dataset Composition

A new demo dataset was processed to benchmark the performance of these components. The dataset consists of 773 documents distributed over five classes, with roughly 150 documents per class. For evaluation purposes, 5% of the documents (approximately 39 files) were randomly selected from the dataset. The classes within the dataset are as follows:

- **Speakers:** 150 documents
- **Microphones:** 171 documents
- **Controller Accessories:** 150 documents
- **Batteries (Non-Rechargeable):** 152 documents
- **Alarms:** 150 documents

Text Evaluation

For the text evaluation, preliminary tests were conducted to determine which PDF text extraction library would provide the best results. The primary concerns during this evaluation were:

- **Accurate Translation:** Ensuring that characters and words are transcribed correctly from the PDF.
- **Accurate Formatting:** Maintaining as much of the original layout (e.g., columns) as possible.

The evaluation of transcription accuracy revealed that approximately 15% of the files contained errors when converting from PDF to text using the selected method.

Library Comparisons:

- **PDFPlumber:** Generally provided acceptable results; however, it struggled with certain complexities inherent in some PDF layouts.
- **PyPDF2:** Performed the worst overall, with a high frequency of errors that compromised the quality of the extracted text.
- **MuPDF:** Produced better results than PDFPlumber, particularly in reading columns, though it still lagged behind in overall character accuracy.

Since our approach utilizes a Bag of Words (BoW) technique and does not require stringent sentence-level semantic coherence (as in BERT-based methods), maintaining perfect sentence structure is less critical. Instead, the priority is to capture the comprehensive list of words that appear in the documents. For this reason, pdfplumber was chosen for character recognition.

Evaluating a representative subset of the data, the results appear consistent across various formatting types and text sizes, with only minor overall inaccuracies. Since the performance is ultimately judged subjectively by the author's review rather than an objective metric, no definitive numerical score can be provided. Additionally, many of the extraneous symbols and formatting issues (such as bullet point artifacts) are subsequently removed during the NLP processing stages.

Limitations

One inherent limitation of the PDF-to-text conversion is that the libraries attempt to recognize any character within the document and then output this to text. Consequently, tables embedded within PDFs are also converted directly into text. However, this is considered acceptable since subsequent NLP techniques are applied to random data points, and full semantic integrity of the original table layout is not essential for our application.

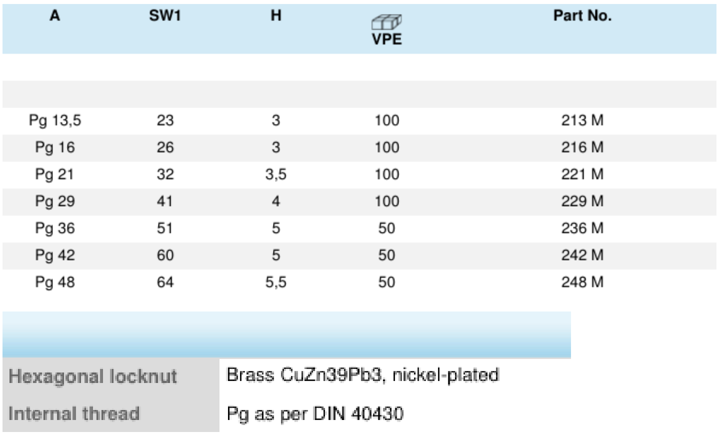
Due to computational constraints, pdfs are limited to 15 pages, as we encountered huge catalogs with hundreds of pages that are infeasible to translate.

Table-to-Text Evaluation

The table-to-text component was integrated into the text conversion workflow primarily for code manageability and does not significantly affect the overall results. There are two types of table evaluations, depending on the table's size and format.

1. Traditional Tables:
These are conventional tables containing data, and the conversion process adheres to expected standards.

2. "Sentences" Tables:
These tables are defined by their limited dimensions—typically no more than five columns or rows. This constraint stems from the underlying table-to-text LLM (TableGPT), which imposes specific formatting limitations.



A	SW1	H	VPE	Part No.
Pg 13,5	23	3	100	213 M
Pg 16	26	3	100	216 M
Pg 21	32	3,5	100	221 M
Pg 29	41	4	100	229 M
Pg 36	51	5	50	236 M
Pg 42	60	5	50	242 M
Pg 48	64	5,5	50	248 M

Hexagonal locknut	Brass CuZn39Pb3, nickel-plated
Internal thread	Pg as per DIN 40430

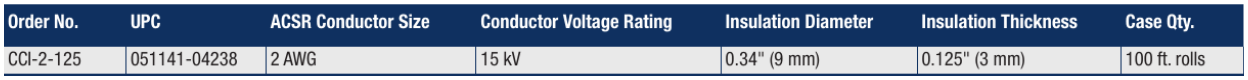
Fig 1: Traditional and sentence table examples

Different LLMs are employed for processing each type of table. For larger tables, a specialized table description method is used, while smaller tables are converted using a more general-purpose LLM that generates a sentence summarizing the data. For the purposes of this report, both approaches are evaluated collectively.

Overall, the translation from tables to text is quite effective. The generated content is context-specific, offering a level of detail that might surpass what a human annotator could produce without additional context. However, some errors do occur—typically due to LLM hallucinations where non-related data is inadvertently introduced.

For instance, the part number "238 PANPT/G" is incorrectly translated into:

- **Temperature Rating:** 238°F
- **Thread Type Identifier:** PANPT/G



Order No.	UPC	ACSR Conductor Size	Conductor Voltage Rating	Insulation Diameter	Insulation Thickness	Case Qty.
CCI-2-125	051141-04238	2 AWG	15 kV	0.34" (9 mm)	0.125" (3 mm)	100 ft. rolls

Moreover, the system accurately retains abbreviations such as "ACSR (Aluminum Conductor Steel Reinforced) Conductor Size" but occasionally introduces minor mistakes, such as fields noting:

- **Conductor Voltage Rating:** Not specified in the given data
- **Insulation Diameter:** Not specified in the given data
- **Insulation Thickness:** Not specified in the given data
- **Case Quantity:** Not specified in the given data

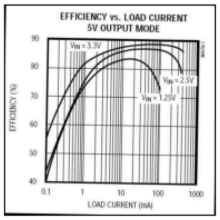
These errors are primarily attributed not to the table-to-text LLM but to limitations in the OCR process used for reading tables, which is the same library employed for text extraction.

Limitations

The developers of TableGPT acknowledge that the table processing module requires significant computational resources and time. As a result, this component of the translation workflow has not been activated for the full demo dataset. Instead of processing all 36 documents, we are currently evaluating only 10 documents, each containing approximately 5 tables.

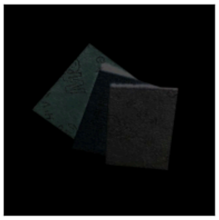
Image-to-Text Evaluation

The process consists of two primary steps. First, a classifier is employed to filter images to ensure that logos and incorrectly cropped images are excluded. This filtering reduces the load on the LLM responsible for image-to-text translation and minimizes noise in the results. The classifier attains approximately 95% accuracy, with occasional misclassifications occurring when it encounters uncommon images (for instance, mistaking a long cable for a banner). This classifier is intentionally designed to be less restrictive—prioritizing the inclusion of more data, even if it introduces a bit of noise.



Product (0)

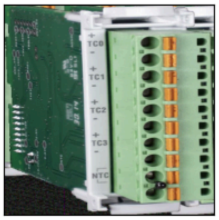
Product (0)



not_product (1)

not_product (1)

not_product (1)



Product (0)

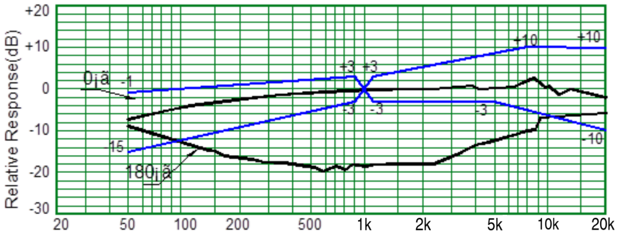
not_product (1)

not_product (1)

Product (0)

Diagrams

When it comes to diagrams, the evaluation of image-to-text conversion becomes more complex due to the inherently ambiguous nature of visual content. For example, some diagrams may lack sufficient context, raising the question of whether including such images is beneficial or if they simply contribute unwanted noise. In contrast, other diagrams are readily interpretable by LLMs and do not require a highly restrictive filtering process.



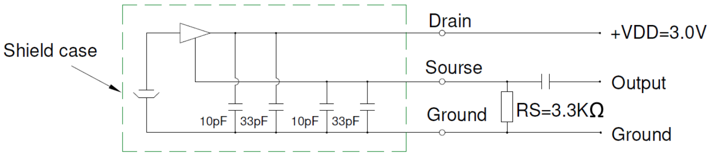
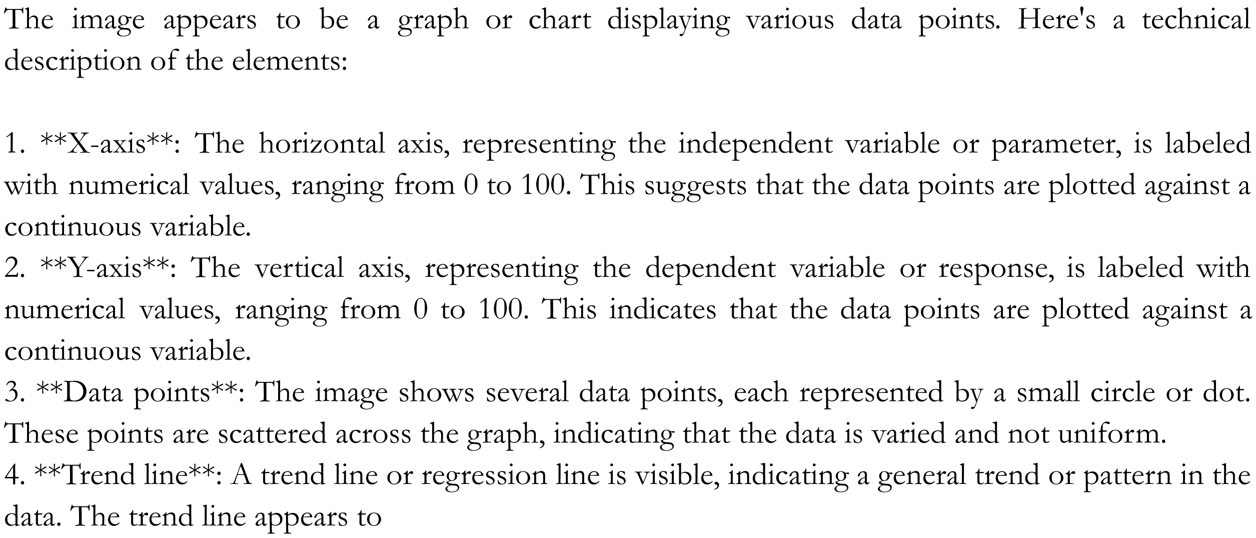
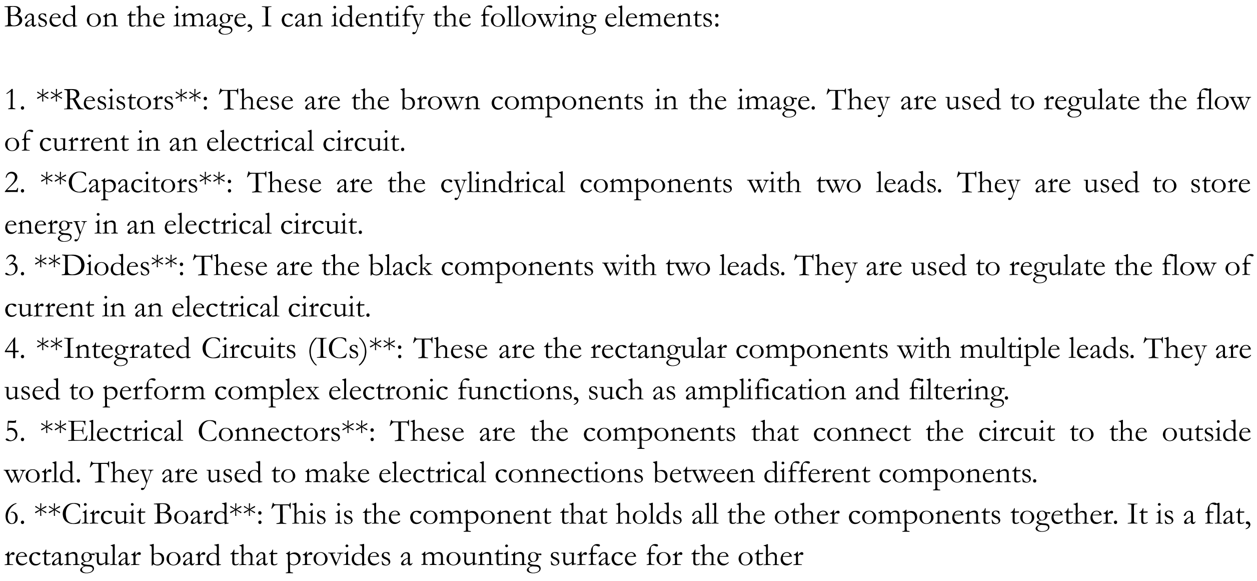


Fig1 : Two different graphs or charts and their respective descriptions



The image appears to be a graph or chart displaying various data points. Here's a technical description of the elements:

1. **X-axis**: The horizontal axis, representing the independent variable or parameter, is labeled with numerical values, ranging from 0 to 100. This suggests that the data points are plotted against a continuous variable.
2. **Y-axis**: The vertical axis, representing the dependent variable or response, is labeled with numerical values, ranging from 0 to 100. This indicates that the data points are plotted against a continuous variable.
3. **Data points**: The image shows several data points, each represented by a small circle or dot. These points are scattered across the graph, indicating that the data is varied and not uniform.
4. **Trend line**: A trend line or regression line is visible, indicating a general trend or pattern in the data. The trend line appears to

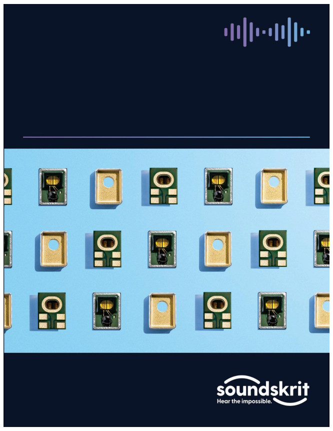


Based on the image, I can identify the following elements:

1. **Resistors**: These are the brown components in the image. They are used to regulate the flow of current in an electrical circuit.
2. **Capacitors**: These are the cylindrical components with two leads. They are used to store energy in an electrical circuit.
3. **Diodes**: These are the black components with two leads. They are used to regulate the flow of current in an electrical circuit.
4. **Integrated Circuits (ICs)**: These are the rectangular components with multiple leads. They are used to perform complex electronic functions, such as amplification and filtering.
5. **Electrical Connectors**: These are the components that connect the circuit to the outside world. They are used to make electrical connections between different components.
6. **Circuit Board**: This is the component that holds all the other components together. It is a flat, rectangular board that provides a mounting surface for the other

Some issues persist, such as occasional hallucinations where the LLM generates incorrect information. Despite these challenges, the overall value of the image-to-text descriptions remains clear, as they effectively translate visual information into a textual format. The most promising area for improvement lies in the fine-tuning of the classifier to better control the flow of images into the translation stage. When the classifier is set to be less restrictive, a higher volume of atypical images may pass through, which in turn can lead to a greater incidence of hallucinations—an issue which is intrinsic to the architecture.

Example of image that could be filtered by fine-tuning the classifier

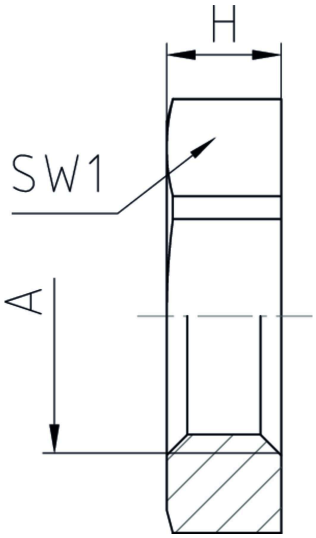


The image appears to be a photograph of a person's face. The elements in the image can be described as follows :

* The face is the primary subject of the image. It is a human face with a mix of facial features, including eyes, nose, and mouth.
* The eyes are the two circular features that are located on either side of the nose. They are a dark brown color and appear to be looking directly at the camera.
* The nose is a small, rounded feature that is located between the eyes. It is a light brown color and appears to be slightly upturned.
* The mouth is a small, curved feature that is located below the nose. It is a light brown color and appears to be slightly upturned.
* The skin is a light brown color and appears to be smooth and even.
* The hair is a dark brown color and appears to be styled in a messy, piecey manner.
* The background is a blurred, out-of-focus area that appears to

---

Example of hallucination on otherwise correct description



The image shows a metal component with a hexagonal shape and a flat top surface. The component has a diameter of approximately 20mm and a height of around 10mm. The surface finish appears to be a medium-coarse finish, with visible machining marks and some minor scratches.

Considerations:

Different visual LLMs were evaluated throughout the process. For a direct comparison of the models tested for image processing, please refer to I*mage Evaluation.* Similarly, for a comparison of the table-related models, see the *Table Evaluation* document.