

## **Cisco Final Report:**

MLDS Team: Andrew, Kelvin, Xin, Xingyu, Ziqiao

### **Report Outline:**

- Problem Definition
- Data Overview
- Solution Overview
- Methodology Overview — Generative AI
- Solution Workflow
  - PDF Extraction
  - RAG
  - Prompt Engineering
- Experiments
  - Initial Results
  - Why we chose RAG
  - Different types of LLMs
  - GPT and Prompt Types
- Results
- Challenge Scenario - Future Results
- Demo
- Next Steps and Recommendations

## **1. Problem Definition**

The project addresses a significant challenge faced by Cisco in managing its supplier product specification files. These files are currently unstructured, meaning the product attributes they contain are not formatted or organized consistently across documents. This lack of structure presents a considerable hurdle, as the current process of data extraction from these files is heavily manual and thus, extremely time-consuming.

The primary objective of this project is to develop an automated data extraction pipeline. This pipeline aims to efficiently collect the necessary information from unstructured document sources, specifically focusing on text-rich PDF files obtained from various suppliers. A critical aspect of this pipeline is to ensure high accuracy in the extraction process, along with the flexibility to adapt to different types of file formats.

Additionally, there's a significant emphasis on user experience. The project plans to build a user-friendly interface that can be easily navigated and utilized by individuals without technical expertise. This interface will facilitate easier access and manipulation of the extracted data, aligning with the project's goal of streamlining the data extraction process and making it more efficient. The project, thus, centers on enhancing the extraction of key attributes from supplier documents, transforming a cumbersome manual process into an automated, accurate, and user-friendly system.

## **2. Data Overview**

Our primary objective is to extract a set of 10 product-related attributes from Cisco's supplier documents. These attributes include Supplier Name, Product Type, Dimensions, Orientation (if applicable), Current Rating, Voltage, Frequency, Impedance, Capacitance, and Temperature ranges. These were identified as the most comprehensive and common attributes across various product types and suppliers, following in-depth consultations with business clients. The specific attributes extracted will vary depending on the product type in question, ensuring a tailored approach to information extraction that aligns with the diverse range of products.

We carefully chose 9 PDF documents representing a diverse array of vendors, products, and document structures. Our selection criteria focused on the variation in document formats, content richness, product categories, and suppliers to guarantee an extensive evaluation across distinct document types. Furthermore, for reasons of

confidentiality and to maintain the integrity of our experimental framework, all selected products fall within the commodity category.

### **3. Methodology Overview**

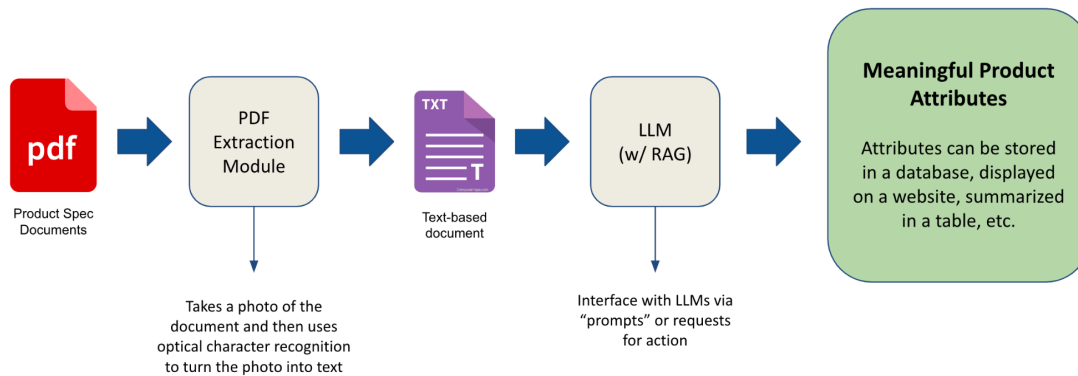
Traditionally, the problem of extracting data from unstructured supplier product specification files could have been approached using a rule-based or pattern matching method, such as Regular Expressions (RegEx). While effective, this approach necessitates custom tailoring for each manufacturer and document type, limiting its flexibility and scalability. Alternatively, machine learning methods like Support Vector Machines (SVM), Convolutional Neural Networks (CNN), or Long Short-Term Memory networks (LSTM) could offer more generalizability. However, these techniques would require an extensive effort in creating a large and accurately labeled training dataset to teach the model the required ground truths.

In contrast, Generative AI (GenAI) is recommended for this use case due to recent advancements in Natural Language Processing (NLP), especially with transformer-based models like GPT-4. These models have significantly improved capabilities in understanding and generating human-like text. Their enhanced ability to comprehend context is crucial for accurately extracting relevant information from various document types.

The advantages of using GenAI are notable. Firstly, these models exhibit adaptability and generalization capabilities superior to rule-based systems and can generalize better than traditional machine learning models. Secondly, GenAI models have a sophisticated understanding of context, enabling them to grasp nuances in language and document structure that rule-based systems might overlook. Additionally, newer GenAI models are multimodal, meaning they can process and interpret both text and images, expanding their applicability.

However, there are significant disadvantages to consider. GenAI models are complex and often require substantial computational resources, which could pose challenges in deployment and maintenance. Moreover, these models are frequently criticized for their lack of interpretability and explainability, often described as "black boxes." This makes it challenging to understand the reasoning behind a model's decision, which can be crucial in certain applications. Despite these drawbacks, the potential benefits of GenAI in terms of adaptability, contextual understanding, and multimodal capabilities make it a compelling choice for this project.

## 4. High-Level Solution Workflow



The flowchart above outlines the proposed solution for extracting meaningful product attributes from unstructured product specification documents. The process begins with the product spec documents in PDF format. The first step involves a PDF Extraction Module that takes a photo of the document and then applies optical character recognition (OCR) to convert the photo into a text-based document. This text document is then fed into a Large Language Model (LLM) with Retriever-Augmented Generation (RAG) capabilities. The LLM interfaces with users through "prompts" or requests for action, leveraging its advanced language understanding to interpret and act upon these prompts.

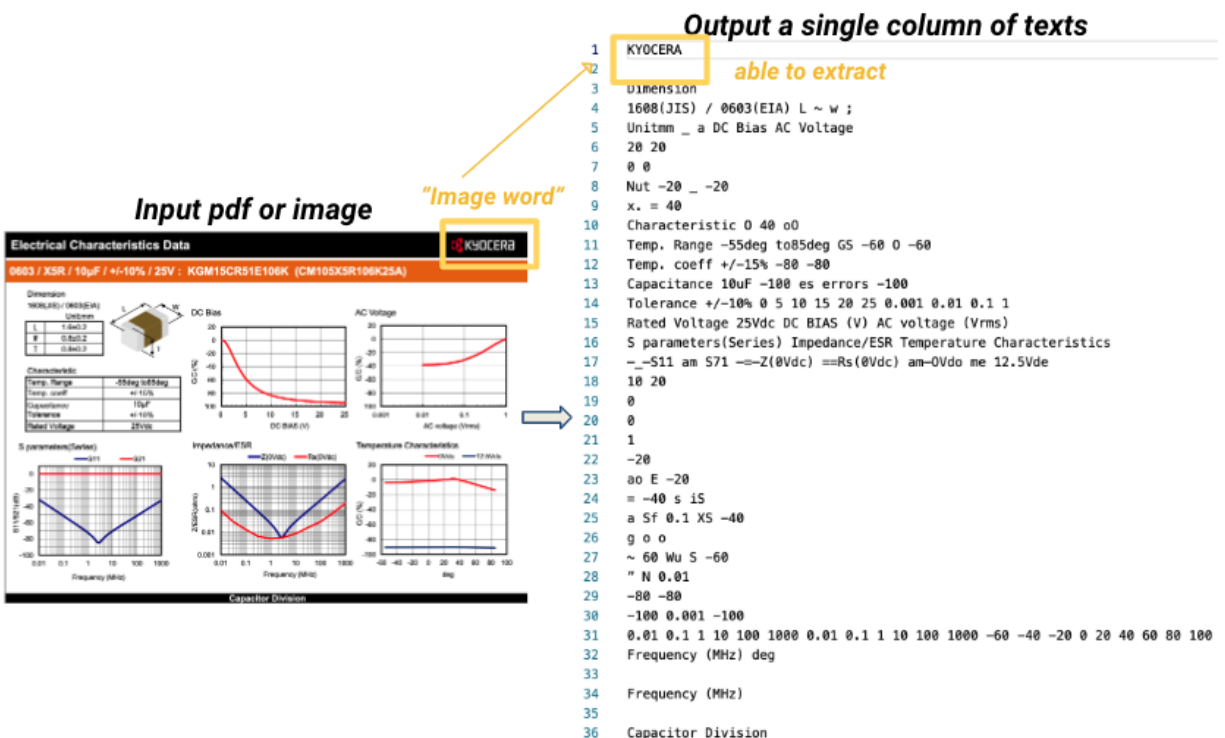
Finally, the output of this process is the extraction of meaningful product attributes. These attributes can then be stored in a database, displayed on a website, or summarized in a table, among other possibilities. This automated flow is designed to streamline the process of data extraction from unstructured documents, making it more efficient and accurate.

## 5. PDF Extraction

PDF extraction involves the retrieval of text, images, and other data from PDF documents. This technique transforms content from a non-editable format into a usable, editable one, catering to a wide range of applications. A pivotal tool in this process is Optical Character Recognition (OCR), with Tesseract being a prominent example. Originally developed by Hewlett-Packard and later enhanced by Google, Tesseract OCR specializes in extracting printed or handwritten text from images. Its accuracy and adaptability make it a preferred choice in various text extraction scenarios.

In our final model, the extraction process begins with Pdf2Image, a tool used to convert PDF files into image formats. This conversion is crucial as OCR technology, like Tesseract, requires image-based input to function effectively. Once the PDF content is converted into images, Tesseract OCR comes into play. It meticulously scans these images to detect and extract text, converting it into a digital format that can be edited and analyzed. This two-step process, involving initial image conversion followed by OCR-based text extraction, is not only efficient but also ensures a high level of accuracy, making it highly suitable for extracting information from various types of PDF documents.

The following image gives an example of how Tesseract works on a sample pdf. Basically, we input the pdf, the model will output a single column of texts for further use. The yellow frame highlights a “image word” in the pdf (image format), which is the supplier name in this case, and Tesseract is able to detect and extract the text in this image word successfully.



This table summarizes all the pdf extraction techniques used. The exploration of various PDF extraction techniques reveals distinct advantages and limitations inherent to each method. The combination of pdf2image and OCR-Tesseract stands out due to its open-source nature and cost-effectiveness, allowing for local usage without financial burden. This method boasts good accuracy and versatility, particularly notable in its ability to interpret "image words" — text embedded within images. However, it requires

careful image preprocessing for optimal results and tends to output text in a single column, not preserving the original document's layout.

Alternatives like Pdfminer/Pdf Plumber and PyPDF (including PyPDF2 & PDFLoader) offer their own sets of pros and cons. Pdfminer/Pdf Plumber is simple and robust, effectively preserving control characters like "\n", but it falls short in reading image-embedded text. PyPDF variants provide compatibility with various platforms (LangChain for PDFLoader, LLaMa for PyPDF2), but they struggle with accurately interpreting certain symbols and also cannot process "image words". GPT-4.0's image reading capability presents high accuracy and versatility, yet its inaccessibility due to a lack of a public API and associated costs is a significant drawback.

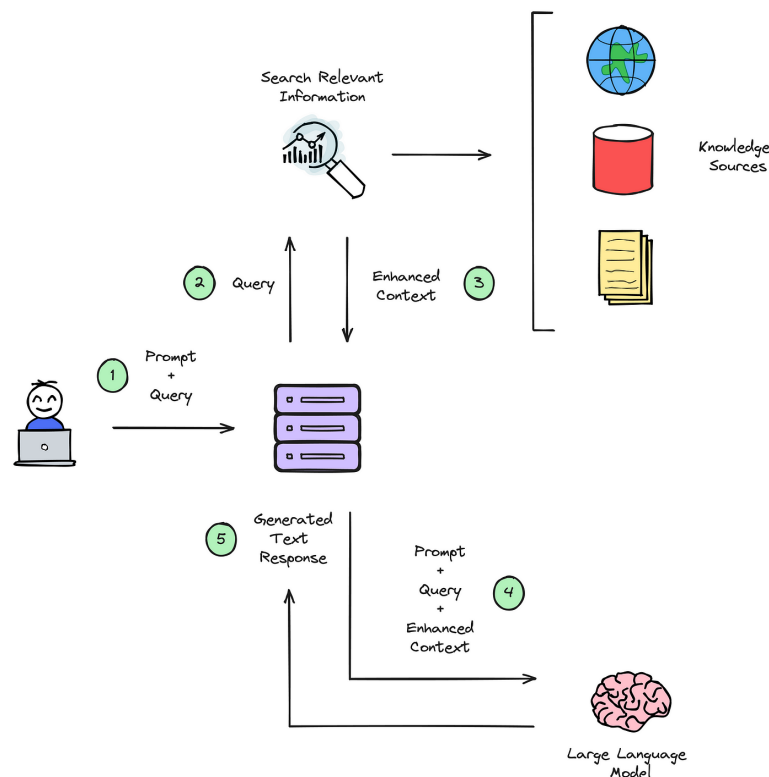
Given these evaluations, the decision to utilize the pdf2image + OCR-Tesseract model is primarily driven by its balance of accuracy, versatility, and cost-effectiveness. The ability to process image words, a crucial requirement for comprehensive text extraction from diverse PDF formats, positions this combination as the most suitable choice. While it does have limitations in terms of text layout preservation and the need for preprocessing, these are outweighed by its overall utility and accessibility, making it an ideal choice for effective PDF data extraction.

Techniques	Pros	Cons
<b>pdf2image + OCR-Tesseract</b>	<ul style="list-style-type: none"><li>● Open-source and free (can be used locally)</li><li>● Good accuracy and versatility</li><li>● Can read "image words"</li></ul>	<ul style="list-style-type: none"><li>● Requires image preprocessing to achieve optimal results</li><li>● Outputs a single column of text, does not preserve the structure</li></ul>
<b>Pdfminer/Pdf Plumber</b>	<ul style="list-style-type: none"><li>● Simple and robust; preserves control characters such as "\n"</li></ul>	<ul style="list-style-type: none"><li>● Cannot read the "image words" such as vendor name</li></ul>
<b>PyPDF (PyPDF2&amp; PDFLoader)</b>	<ul style="list-style-type: none"><li>● PDFLoader has more compatibility with LangChain and so does PyPDF2 with LLaMa</li></ul>	<ul style="list-style-type: none"><li>● Some signs cannot be read correctly. E.g. Read °C as oC</li><li>● Cannot read "image words"</li></ul>

<b>GPT4.0-image reading</b>	<ul style="list-style-type: none"> <li>● Good accuracy and versatility</li> <li>● Can read “image words”</li> </ul>	<ul style="list-style-type: none"> <li>● No accessible API, not free</li> </ul>
-----------------------------	---	---

## 6. Retrieval Augmented Generation (RAG)

Retrieval Augmented Generation (RAG) is a technique that substantially enhances the performance of LLMs by integrating them with external knowledge sources. The primary goal of RAG is to contextualize the information accessible to a language model, thereby elevating the relevance and precision of the text it generates. This is especially beneficial when it comes to specific facts or contemporary data that the LLM may not have encountered during its training phase.



The figure above illustrates how RAG works with LLMs:

- **1. Prompt + Query:** The process begins with the user inputting a prompt that includes a query. This query is a statement or question that encapsulates the user's need for information or the topic they are interested in exploring.
- **2. Query:** For each user query, the system performs a similarity search against a set of external knowledge sources. These sources can be databases, the internet, or any structured repository of information that the system has access to.
- **3. Enhanced Context:** The system retrieves contextually relevant information from these knowledge sources that closely matches the user's query (a common

metric used is cosine similarity between query and knowledge sources content). This step is crucial as it ensures that the context used to generate the response is as pertinent as possible to the user's request.

- **4. Prompt + Query + Enhanced Text:** The extracted context is then synthesized with the user's original prompt and query. This combined input is fed into the LLM, providing it with a richer data source to draw from when generating a response.
- **5. Generated Text Response:** Finally, the LLM processes this integrated input and generates a text response. Because the LLM now has access to enhanced context, it can provide answers that are not only based on its pre-existing knowledge but also on the most relevant and recent information retrieved from external sources.

Utilizing RAG enables LLMs to generate responses that are significantly more precise and contextually appropriate. In this project, this implies that each user query is enhanced with specific supplier document context fed into the LLM. Therefore, the outputs are derived from uploaded documents, ensuring that the responses are specifically tailored, rather than being limited to the model's training data. With this strategy, Cisco is positioned to harness AI-driven solutions that not only respond to immediate needs but also bring a host of additional advantages.

Building upon the core benefits already outlined, RAG presents a suite of further advantages:

- **Efficient Processing of Complex Documents:** The ability of RAG to segment complex documents into manageable parts allows for an in-depth analysis of extensive materials, overcoming typical processing constraints.
- **Transparency with Source Text:** Providing source texts with responses fosters an environment of trust and accountability, which is paramount in business settings where verification and compliance are critical.
- **Versatility with Information Sources:** The flexibility of RAG to draw from a diverse array of information sources ensures that the LLM's outputs are well-rounded and reflective of the most current data available.

While the transition to RAG involves consideration of factors such as latency and information completeness, the advantages it brings—specialization, thoroughness, transparency, and adaptability—are transformative for Cisco seeking to enhance their document intelligence and information extraction processes.

## **7. Prompt Engineering**

Prompt Engineering forms a crucial component in the high-level solution workflow for our project at Cisco. This technique involves the strategic formulation and optimization



of input prompts to effectively steer the Generative AI (GenAI), such as Large Language Models (LLMs), to generate accurate and relevant outputs.

The implementation of prompt engineering significantly enhances the effectiveness of our data extraction process from supplier product specification files. Key aspects of our approach to prompt engineering include:

#### Enhanced Accuracy

- Detailed Explanations: We modify the prompts to encompass detailed explanations for each product attribute. This helps in clarifying the context and scope of the information required from the LLM.
- Focused Queries: To ensure comprehensive and relevant responses, we design prompts to query the LLM individually for each attribute. This avoids overwhelming the model with multiple attribute queries in a single prompt and helps in maintaining focus and relevance in the responses.
- Comprehensive Yet Concise: The prompts are engineered to guide the LLM to provide thorough answers covering all pertinent aspects of an attribute, while simultaneously avoiding unnecessary repetition or verbosity.

#### Avoiding Hallucination

- Clear Instructions for Uncertainty: A critical part of our prompt strategy includes embedding instructions within the prompts, directing the LLM to respond with phrases like “I don’t know” in cases of uncertainty. This approach significantly reduces the likelihood of the model 'hallucinating' or providing incorrect information when it is not confident about the answer.

#### Ensuring Correct Format

- JSON Formatting Guidance: We instruct the LLM to structure its responses in a specific format, such as JSON. This is complemented by providing the model with examples of the desired output structure.
- Nested Dictionaries for Multiple Values: For attributes that encompass multiple values, the prompts are designed to encourage the model to generate responses in the form of nested dictionaries. This maintains clarity and organization in the extracted data.

In summary, our approach to prompt engineering is a balanced act of refining and iterating prompts to improve the performance of the LLM in extracting data from supplier documents. While this process demands meticulous effort and consideration, especially in managing engineering effort associated with longer prompts, the art of prompt engineering, plays a pivotal role in the successful deployment of our solution, aligning

with Cisco's vision of streamlining data extraction processes while ensuring high standards of accuracy and user experience.

## 8. Experiments

### Initial Results

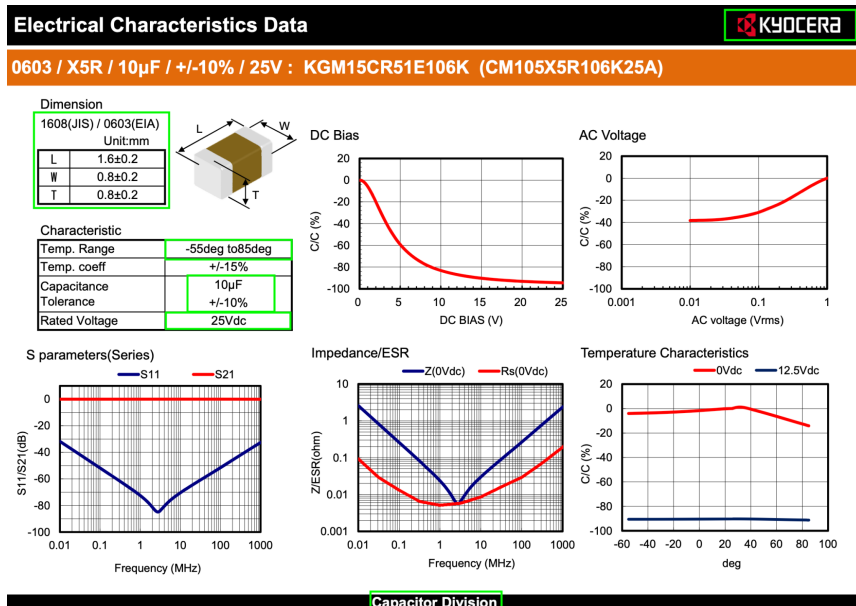
To evaluate model performance, we compare the model output with ground truth values appeared in the pdf. The evaluated pdfs are divided into two scenarios, First scenario is when documents are short and have clear tables, and the other scenario is when documents are very long and have complicated tables and plots.

### Scenario 1: When PDFs are short with simple layouts

We discovered that the GPT-3.5 model can perform pretty well on PDF documents that are short and have simple layouts. The model is able to achieve accuracy near 100%, and thus this is the scenario where no manual proofing is needed for approval.

#### Example document 1:

The screenshot presented illustrates the layout of the first document subjected to our evaluation. This document is composed of a single page and features simplistic tables with attribute names explicitly stated. Boxes highlighted in the screenshot indicate where the model has to refer to in the document in order to identify correct information for each attribute.



In the comparison table provided, values highlighted in green signify that the model has accurately identified the relevant information for the attributes. According to this table, the model achieved 100% accuracy in extracting data from this document.

PDF Name	Answer Type	Supplier Name	Product Type	Dimension	Voltage	Capacitance	Temperature
<a href="#">KGM15CR51E106K-DATA.pdf</a>	Model Output Ground True Value	KYOCERA KYOCERA	Capacitor Capacitor	1608(JIS) / 0603(EIA) L ~ w 1608(JIS) / 0603(EIA) L ~ w	25Vdc 25V	10uF 10 uF	-55deg to 85deg -55C to 85 C

### Example document 2:

The second document further demonstrates the model's good performance on this kind of document. The result is still almost 100% accurate except for one parameter with extra width

Lelon P/N : VZH680M2ATR-1313S		LELON ELECTRONICS CORP. VZH 68 $\mu$ F / 100 V – 12.5 $\phi$ × 13.5L		Page : 1 / 1																												
CUSTOMER : CUSTOMER P/N:																																
<p>DIAGRAM OF DIMENSIONS</p> <table border="1"> <thead> <tr> <th>Unit: mm</th> </tr> </thead> <tbody> <tr><td>φD</td><td>12.5</td></tr> <tr><td>L</td><td>13.5 ± 0.5</td></tr> <tr><td>A</td><td>13.0</td></tr> <tr><td>B</td><td>13.0</td></tr> <tr><td>C</td><td>13.7</td></tr> <tr><td>W</td><td>1.1 ~ 1.4</td></tr> <tr><td>P</td><td>4.4 ± 0.2</td></tr> </tbody> </table>					Unit: mm	φD	12.5	L	13.5 ± 0.5	A	13.0	B	13.0	C	13.7	W	1.1 ~ 1.4	P	4.4 ± 0.2													
Unit: mm																																
φD	12.5																															
L	13.5 ± 0.5																															
A	13.0																															
B	13.0																															
C	13.7																															
W	1.1 ~ 1.4																															
P	4.4 ± 0.2																															
Items		Performance																														
Category Temperature Range		-55°C ~ +105°C																														
Capacitance Tolerance		-20 % ~ +20 % (120 Hz, 20°C)																														
Surge Voltage		125 VDC																														
Leakage Current		I ≤ 68 $\mu$ A After 2 minutes																														
Dissipation Factor (Tan $\delta$ )		≤ 0.07 (120 Hz, 20°C)																														
Impedance		< 0.32 $\Omega$ (100kHz, 20°C)																														
Ripple Current (rms)		450 mA (100kHz, 105°C)																														
Low Temperature Characteristics (120 Hz)		<table border="1"> <tr> <td>Z(-25°C) / Z(+20°C)</td> <td>2</td> </tr> <tr> <td>Z(-55°C) / Z(+20°C)</td> <td>3</td> </tr> </table>			Z(-25°C) / Z(+20°C)	2	Z(-55°C) / Z(+20°C)	3																								
Z(-25°C) / Z(+20°C)	2																															
Z(-55°C) / Z(+20°C)	3																															
Ripple Current & Frequency Multipliers		<table border="1"> <thead> <tr> <th>Frequency (Hz)</th> <th>50, 60</th> <th>120</th> <th>1k</th> <th>10k up</th> </tr> </thead> <tbody> <tr> <td>Multiplier</td> <td>0.60</td> <td>0.70</td> <td>0.85</td> <td>1.00</td> </tr> </tbody> </table>			Frequency (Hz)	50, 60	120	1k	10k up	Multiplier	0.60	0.70	0.85	1.00																		
Frequency (Hz)	50, 60	120	1k	10k up																												
Multiplier	0.60	0.70	0.85	1.00																												
Life Test:		<table border="1"> <tr> <td>Capacitance Change</td> <td>Within ±30 % of initial value</td> </tr> <tr> <td>Dissipation factor</td> <td>Less than 300% of specified value</td> </tr> <tr> <td>Leakage Current</td> <td>Within specified value</td> </tr> </table>			Capacitance Change	Within ±30 % of initial value	Dissipation factor	Less than 300% of specified value	Leakage Current	Within specified value																						
Capacitance Change	Within ±30 % of initial value																															
Dissipation factor	Less than 300% of specified value																															
Leakage Current	Within specified value																															
Endurance: After 5000 Hrs at 105°C																																
Shelf Life Test: After 1000 Hrs at 105°C																																
Standards		JIS C 5101-1, -18																														
Remarks		RoHS Compliance & Halogen-free																														
<p>Marking: Each capacitor shall be marked with the following information.</p> <p>A 3 → January, 2013</p> <p>→ The suffix of A. D. → Month of manufacture</p> <table border="1"> <thead> <tr> <th>Month</th> <th>1</th> <th>2</th> <th>3</th> <th>4</th> <th>5</th> <th>6</th> </tr> </thead> <tbody> <tr> <td>Code</td> <td>A</td> <td>B</td> <td>C</td> <td>D</td> <td>E</td> <td>F</td> </tr> <tr> <th>Month</th> <th>7</th> <th>8</th> <th>9</th> <th>10</th> <th>11</th> <th>12</th> </tr> <tr> <td>Code</td> <td>G</td> <td>H</td> <td>I</td> <td>J</td> <td>K</td> <td>L</td> </tr> </tbody> </table>					Month	1	2	3	4	5	6	Code	A	B	C	D	E	F	Month	7	8	9	10	11	12	Code	G	H	I	J	K	L
Month	1	2	3	4	5	6																										
Code	A	B	C	D	E	F																										
Month	7	8	9	10	11	12																										
Code	G	H	I	J	K	L																										
<p>Marking color: Black</p> <p>* Please refer to "Precautions and Guidelines for Aluminum Electrolytic Capacitors" of Lelon's catalog.</p>																																
<p>Publish Date: April 12, 2013</p> <p>Revise Date:</p> <p>Edition No.: 1</p>	<p>Approval Signatures:</p> <p>Please return one copy with your approval</p>		<p>Approved</p> <p>研發部 APR. 12, 2013 林永洲</p>	<p>Checked</p> <p>研發部 APR. 12, 2013 黃建智</p>																												
<p>Designed</p> <p>研發部 APR. 12, 2013 熊金華</p>																																

### Comparison table:

PDF Name	Answer Type	Supplier Name	Product Type	Dimension
<a href="#">Vzh680m2atr-1313s.pdf</a>	Model Output Ground True Value	LELON ELECTRONICS LELON ELECTRONICS	Capacitor Capacitor	Length: '12.5 mm', 'Width': '10 mm', 'Height': '13.5 mm' Length: 12.5 mm, Height: 13.5mm


PDF Name	Answer Type	Voltage	Impedance	Capacitance	Temperature
<a href="#">Vzh680m2atr-1313s.pdf</a>	Model Output Ground True Value	100 V 2A (100 V)	< 0.320 (100kHz, 20°C) < 0.320 (100kHz, 20°C)	68 uF 68 uF	-55°C ~ +105°C -55C to 105C

### Scenario 2: When PDFs are long with complex layouts

For documents extremely long and with many complex tables and plots, the model has less optimal performance than in the first scenario, and this is the case where manual check is required to validate results.

Example document 3:

Although document 3 has only two pages, it has very complex formattings, which could impact on the model's ability to grasp the right information.




KOA SPEER ELECTRONICS, INC.

# RK73H

precision 0.50%, 1% tolerance  
thick film chip resistor

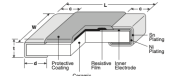
RoHS  
COMPLIANT



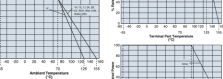
## features

- Products with lead-free terminations meet EU RoHS requirements. EU RoHS regulation is not intended for Pb-glass contained in electrode, resistor element and glass.
- AEC-Q200 Tested: 2001 (H), 2042 (E), 0603 (J), 0608 (2A), 1206 (2B), 1210 (2C), 2017 (2H-W2H), 2512 (3A/3W/3A2C)

## dimensions and construction



## Derating Curve



For resistors operating at an ambient temperature of 70°C or higher, the power shall be derated in accordance with the above derating curve.

When the terminal part temperature of the resistor exceeds the rated temperature shown above, the power shall be derated according to the derating curve. Please refer to "Reduction of the derating curve based on the terminal part temperature" on the beginning of our catalog below for details.

\*Power/dissipation (EA) package size codes.

\*\* RK73H 3A and 3A2 are also not available (different "P" dimensions = 0.4 ± 0.04 mm)

## ordering information

RK73H		T		Packaging		1003	F							
Type	Size	Characteristics	Termination Material	Taping		Nominal Resistance	Tolerance							
1F	1F, 1J	New A-Heat shock resistance **	G-Au ** (L:Sn/Pb*)	TK: 4mm wide - from pitch package embodied		3 significant figures + 1 multiplier *9 indicates decimal on value ×1000	J: ±0.5% F: ±1%							
2B	2B, 2D			TBL: 2mm pitch press paper **										
2E	2E			TFL: 1.2mm pitch punch paper **										
2A	2A, 2A2			TE: 4mm pitch package embodied										
W3A2	W3A2													
		* With type A only T is available as the terminal surface material. ** With type 1H, 1J, W3A, W3A2 only T is available as the terminal surface material. *** Based on aging specification of 1H to TCM. Previously available "TC" is not recommended for new design.												
		Specifications given herein may be changed at any time without prior notice. Please consult technical specifications for details or order.												

KOA Speer Electronics, Inc. • 199 Blvd. of Freedom • Bradford, PA 16701 • USA • 814-362-5536 • Fax: 814-362-8838 • [www.koeaspec.com](http://www.koeaspec.com)

In the following comparison table, values colored in red indicates that the model does not output the correct answer, and values colored in yellow means that the model found partially correct answers.

PDF Name	Answer Type	Supplier Name	Product Type	Dimension
<a href="#">RK73H.pdf</a>	Model Output Ground True Value	Not found <a href="#">KOA Speer Electronics, Inc.</a>	Resistor <a href="#">Resistor</a>	Length: 0.016 inches (0.4 mm), Width: 0.008 inches <a href="#">1E (0402): 0.04 in X 0.02 in</a>

### How complex formatting can affect model performance?

To grasp how complex formatting impacts the model's performance, let's examine the extraction of the dimension attribute from document three. This document poses a specific challenge: the model must identify the correct data from the second row of the dimension table as shown in the screenshot. Complicating this task is the

presence of 'noise' – such as engineering drawings, line plots, and adjacent text – which can blend with the table's content during the PDF-to-text conversion without using GPT's image API.

Moreover, the structure of the table itself, which includes cells spanning multiple rows, makes the model more difficult to correctly capture the original layout. This could lead to a situation where the correct values are not properly aligned with the targeted keywords, resulting in inaccuracies in the model's output. In these cases, a manual check is required to validate the result.

Example document 4:

Document 4 presents a unique challenge due to its extensive length, encompassing 96 pages. This length can potentially impact the model's performance, as it necessitates the processing and searching through a significantly larger volume of content compared to shorter documents. The increased amount of data requires more extensive scanning to locate relevant sections, which could strain the model's capabilities and affect its efficiency and accuracy.

Comparison Table:

PDF Name	Answer Type	Supplier Name	Product Type	Dimension
<a href="#">iMX8MMIEC.pdf</a>	Model Output Ground True Value	NXP Semiconductors NXP Semiconductors	Not found (App Processor) i.MX 8M M	14 x 14 mm Package Length (mm):14 mm, Package Width (mm):14 mm

**Why We Chose RAG-based Approach**

After a thorough analysis of the available methodologies for PDF data extraction, the decision to adopt the Retrieval-Augmented Generation (RAG) approach was made, primarily for its advanced capabilities in handling specialized and complex data. One of the key strengths of RAG is its ability to be paired with specific domain databases, providing highly relevant and specialized responses. This feature is particularly valuable in fields where accuracy and domain-specific knowledge are critical.

Moreover, RAG's ability to effectively manage longer texts sets it apart. It can split extensive documents into manageable chunks, ensuring comprehensive data processing without the limitations imposed by token count, a common issue in non-RAG methods. This is especially beneficial when dealing with lengthy PDF documents. Additionally, RAG maintains transparency in data processing by providing the source text, an essential feature for verification and validation of the extracted information. While this approach might introduce some latency and there's a risk of potential information loss, the benefits of accurate, domain-specific extraction and handling of complex, lengthy documents far outweigh these concerns. Furthermore, RAG's flexibility

in working with a variety of information sources makes it a versatile tool for diverse PDF extraction needs. Thus, choosing RAG aligns with the objectives of efficient, accurate, and specialized data extraction from PDF documents.

## **Different Types of LLMs**

In the rapidly evolving landscape of language models, the distinction between open-source and closed-source models presents significant implications for their application in various projects. Open-source models like Mistral-7B and Mosaic-mpt7b are publicly accessible, allowing for customization and community-driven improvements. They offer the advantage of being free and adaptable to specific needs. However, these models often lack the extensive support found in closed-source alternatives, require considerable resources for training and deployment, and may underperform compared to their state-of-the-art counterparts.

Closed-source models such as GPT-3.5 and GPT-4.0, developed by organizations like OpenAI, represent the cutting edge in language model technology. They typically offer more advanced capabilities and better performance, along with reliable support and regular updates. However, these benefits come at a cost, including licensing or usage fees, and present limitations in terms of customization due to their proprietary nature.

In the context of our project, the decision to use GPT-3.5 as the final model was driven by a need to balance cost and performance. While GPT-3.5 may not be as advanced or accurate as GPT-4.0, especially for complex tasks, it provides a suitable level of accuracy for our requirements at a more manageable cost. GPT-4.0, offering superior performance for complex tasks, is recommended for scenarios where the budget allows, considering its higher expense relative to GPT-3.5.

Additionally, GPT-4.0-turbo, notable for its ability to directly read images through an API, stands out for handling highly unstructured image information. However, this newer model comes with a higher cost, making it a less viable option for projects with limited budgets. Meanwhile, open-source models like Mistral-7B and Mosaic-mpt7b, despite being free and secure, fall short in performance. Mistral-7B struggles with detecting certain information attributes, and Mosaic-mpt7b faces challenges in following precise instructions and overall performance.

In conclusion, our choice of GPT-3.5 is guided by the need for a reliable and cost-effective solution that adequately meets the project's demands without the higher costs associated with more advanced models like GPT-4.0 and GPT-4.0-turbo. While

open-source models offer financial and customization benefits, their limitations in performance and support render them less suitable for our specific needs.

Models	Pros	Cons	Decision
<b>GPT3.5</b>	<ul style="list-style-type: none"> <li>More accurate than open-source models</li> </ul>	<ul style="list-style-type: none"> <li>Less accurate and advanced than GPT-4.0</li> </ul>	We use as final model for this project considering the balance between cost and performance
<b>GPT4.0</b>	<ul style="list-style-type: none"> <li>More accurate than GPT 3.5 on complex tasks</li> </ul>	<ul style="list-style-type: none"> <li>More expensive than GPT 3.5</li> </ul>	Recommend to use for pdf reading if budget allows
<b>GPT4.0-turbo</b>	<ul style="list-style-type: none"> <li>Can directly read image through API</li> </ul>	<ul style="list-style-type: none"> <li>New model and high cost</li> </ul>	Recommend to use if needs to deal with highly unstructured image information
<b>Mistral-7B</b>	<ul style="list-style-type: none"> <li>Completely free and secure</li> </ul>	<ul style="list-style-type: none"> <li>Unable to detect some information of the attributes</li> </ul>	Not suggested
<b>Mosaic-mp t7b (instruct)</b>	<ul style="list-style-type: none"> <li>Completely free and secure and accepts very long token inputs</li> </ul>	<ul style="list-style-type: none"> <li>Hard to convince the model to follow the exact instruction</li> <li>Bad performance</li> </ul>	Not suggested

## Prompt Types

### Basic prompt:

"I'm looking for information in this document. Here is the information I'm looking for Supplier Name, Product Name, Product Type, Dimensions, Orientation, Current Rating, Voltage, Frequency, Impedance / Capacitance, Temperatures"

- Output Example:

- The supplier name is Amphenol ICC. The product names are EXAMAX2™ and EXAMEZZ2™ Connector System. The product type is Mezzanine connectors.
  - The document does not provide specific information on dimensions and orientation.
  - The operating current rating can be found in the AICC Power Design Guide GS-20-0456.
  - The agency voltage rating is less than 30 VAC RMS / DC and the non-agency voltage rating is 150 Vac RMS or Vpc Maximum.
  - The document does not provide information on frequency and impedance/capacitance. The operating temperature range is -55 to 85 °C.
- Observations:

The output contains a list of attributes and results. Challenges were observed with attributes listed with a slash (e.g., Impedance / Capacitance), affecting the clarity of the final output.

Final prompt:

System message: Use the following pieces of context to answer the question at the end. If you don't know the answer, just say that you don't know, don't try to make up an answer.

Prompt: Please extract and compile all information related to the following attributes from the document, avoiding any duplication. For each attribute, include the specific source or context for each unique value. Present the findings of each attribute in a single dictionary entry.

Exclude any redundant information.

(a) Supplier Name

...

(i) Temperatures(temperatures under any conditions)

The answer should be in the following format:

```
{
  "attribute": "Supplier Name",
  "values": [
    {"value": "unique value 1", "source": "context or source"},
    {"value": "unique value 2", "source": "context or source"}
  ]
}
```



],

"summary": "Provide a general summary or observation about the extracted

- Output example

```
{
  "attribute": "Temperatures",
  "values":
  [{"value": "25 +/-5 degC", "source": "9.2 Inspection Conditions"},
  {"value": "-55 and +85 deg C", "source": "8.1 Thermal Shock"},
  {"value": "85 deg C", "source": "8.3 High Temperature Life"} ],
  "summary": "The document provides information on temperatures under
different conditions."
},
{
  "attribute": "Voltage",
  "values":
  [{"value": "500 volts DC", "source": "6.3 Dielectric Withstanding Voltage"},
  {"value": "500 volts DC", "source": "6.2 Insulation Resistance"}],
  "summary": "..."
}
```

- Observations

The output is structured as JSON with nested dictionaries for each attribute. Each entry contains the attribute name, multiple values with their respective sources, and a summary.

Both GPT-4 and GPT-3.5 successfully produced outputs in the instructed format, providing references such as explanations, section numbers, or section titles.

The inclusion of specific instructions for uncertainty ("If you don't know the answer, just say that you don't know") reduces the likelihood of incorrect or fabricated responses.

The prompt directs the model to avoid duplication and provide context for each value, enhancing the accuracy and relevance of the extracted information.

## Multi-Prompt vs Single-Prompt

In the development of our automated data extraction pipeline, we evaluated two primary prompt strategies: Multi-Prompt and Single-Prompt. Below is a detailed analysis of each, highlighting their respective advantages and limitations:

Prompt	Definition	Pros	Cons
<b>Multi-Prompt</b>	Individual prompt for each attribute and call LLM API separately.	<ul style="list-style-type: none"> <li>• More precise, relevant and comprehensive answers</li> <li>• Error in one response doesn't affect the extraction of other attributes.</li> <li>• Allows for tailored prompts for each attribute, thus handling complex attributes more effectively.</li> </ul>	<ul style="list-style-type: none"> <li>• More API calls, more tokens, higher cost</li> <li>• Sequential processing can be slower</li> </ul>
<b>Single-Prompt</b>	One prompt for all attributes within a document and call API once.	<ul style="list-style-type: none"> <li>• Fewer API calls, fewer tokens, lower cost</li> <li>• Faster especially for less complex document</li> </ul>	<ul style="list-style-type: none"> <li>• Less precise responses</li> <li>• Error in the response can affect the extraction of all attributes.</li> <li>• Limited customization</li> </ul>

Based on our analysis, we recommend the Multi-Prompt Strategy. This approach aligns with our project's emphasis on accuracy, reliability, and the ability to handle complex and diverse document structures.

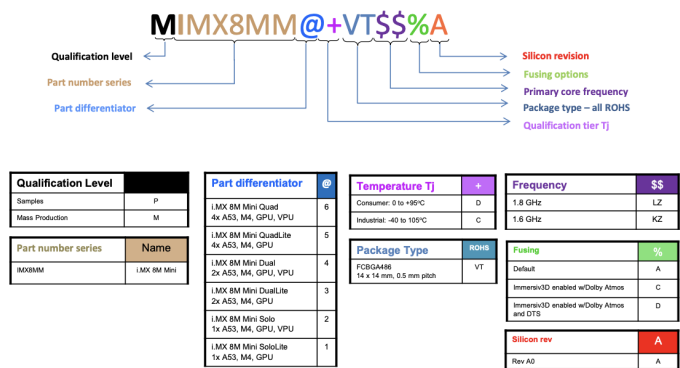
While the Multi-Prompt Strategy entails higher resource usage, these trade-offs are justified by the significant gains in precision, relevance, and the isolated impact of errors. The strategy's ability to tailor prompts for each attribute ensures effective handling of complex and varied data, crucial in accurately extracting information from Cisco's diverse range of supplier documents.

## 9. Challenge Scenario

Initially, the GPT API could only handle text parsing, which meant it was unable to interpret the formatting within documents. However, the image indicates that as of mid-November, the OpenAI API has been upgraded to accommodate pictures, leveraging the multimodal capabilities of ChatGPT 4.0, which can accept images.

Even with these advanced capabilities, Multimodal GPT 4.0 still requires specific prompting to parse data from images correctly. It highlights a case where the model, when prompted to decode the nomenclature from an image, was generally able to summarize tables effectively but with some small inaccuracies, particularly with the Package Type.

The Cisco team pointed out that they have a specific class of document that requires “decoding” - see the picture below. Documents with this format were not conducive to the text-only GPT, but when testing on the image-based GPT, we obtained good results. Furthermore, when prompted to decode a part number that had not been seen before, the model was capable of decoding the part number correctly and listing the attributes associated with it.



Above is an example part name "MIMX8MM@+VT\$S%A" with arrows pointing to a key that explains the meaning of each section of the part name. The key includes terms like "Silicon revision," "Fusing options," "Primary core frequency," "Package type – all RoHS," and "Qualification tier Tj."

The GPT-image based model still required specific prompting to get to the right attributes or to list all known attributes; however, this shows extreme promise over the text-based version of GPT.

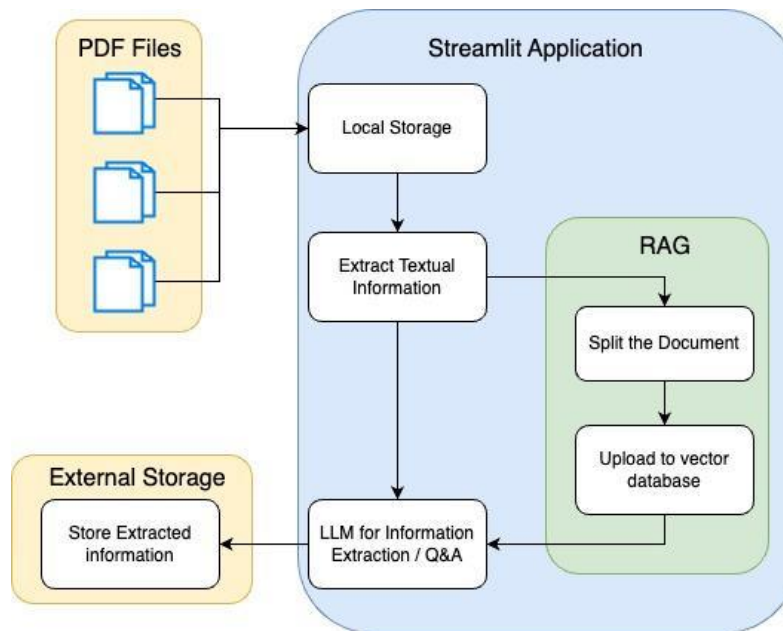
10. Demo

A demonstration of our Retrieval Augmented Generation (RAG) implementation with ChatGPT-4 is available for review at the following link: [Demo Video](#).

This implementation within a Streamlit application demonstrates RAG's capabilities through an intuitive interface, streamlining the document upload process and subsequent information extraction. The system not only accurately extracts essential attributes but also enables users to engage in a dynamic Q&A dialogue with the

content. Responses are not only precise but are also accompanied by source citations, enhancing transparency of the system.

The architecture of our demo, as shown in the figure below, is an efficient pipeline designed for enhanced document processing and interaction. It involves several steps for the document extraction:



1. **Initial Input and Storage:** Documents are uploaded and temporarily stored, priming them for the subsequent Optical Character Recognition (OCR) process.
2. **OCR Processing:** Tesseract OCR technology is used to convert PDFs/images into text.
3. **RAG Processing:** Subsequently, the RAG module processes this text, segmenting documents into smaller sections for thorough analysis. These sections are then indexed in a vector database, optimizing them for quick retrieval.
4. **Information Extraction and Q&A:** The pipeline culminates with a conversational interface that utilizes the indexed information for on-the-spot information retrieval and Q&A engagement.

To satisfy Cisco's need, our demonstration showcases three primary use cases:

1. **Single Document Upload:** When a user uploads a single document, the application extracts key attributes, displaying them in a structured table within the Streamlit interface.
2. **Multiple Document Upload:** The system adeptly handles the upload and processing of multiple documents, extracting and tabulating attributes from each one.
3. **Batch Processing:** For a collection of documents within a folder, the application performs batch processing, extracting attributes from all documents efficiently.

In every case, attributes are presented in a table view, with the option for users to download the data as a CSV file. Concurrently, key attributes are stored in JSON format, ready for further processing.

The Streamlit application is a testament to the effective integration of AI with the document intelligence framework, delivering a robust tool for key information extraction and batch process multiple documents.

## 11. Recommendations and Next Steps

Firstly, we see OpenAI new API to be immensely useful in decoding complex product documents, aiding in simplifying technical information which was previously challenging. Additionally, the API's ability to read and summarize graphs and line charts is a standout feature, enhancing data analysis and reporting. In our tests, we found that the conversion of data tables to JSON format was successful, although it will require prompt tuning for optimal performance.

An important aspect of our approach involves human-in-the-loop validation. For complex or extensive documents, human oversight will be necessary to ensure accuracy, while simpler, shorter documents can be efficiently managed solely by the automated pipeline. This dual approach ensures both precision and efficiency.

Customized prompts, specifically tailored with particular attributes, have shown to be more effective than generic ones. However, this will entail a dedicated effort in prompt engineering to fully harness the potential of the technology. Regarding model selection, given the enhanced performance and extended context window, the GPT-4.0-turbo model seems to be the most suitable choice for future builds.

To determine the most effective prompt, I suggest following our established "Success Framework". This involves evaluating the model's output against the ground truth across a wide array of documents (over 30 is preferred), ensuring a comprehensive assessment of its effectiveness.

Finally, considering the rapidly changing technical environment, I recommend building the final pipeline in a modular fashion. This means keeping the RAG, the model itself, file reading mechanisms, and the web app as independent components. Such a structure allows for easier updates and modifications, ensuring the system remains agile and adaptable to new developments.

Overall, these recommendations are poised to leverage the new API's capabilities effectively, aligning with Cisco's operational needs and future-proofing the technological infrastructure.