An Intelligent Semantic Search System for Digital Product Part Design Catalogues in Manufacturing

Research Questions

Pablo de Vicente Abad, 20/05/2025

1. Research Questions

This is the original proposal for research questions. Moharram expressed some concerns about the first one, so in this report i'll go over it in more detail and rewrite it

1. How can documents / product catalogs = {text,images,tables} be transformed into a vector space
2. Is it more efficient to serialize into single vector or do multivector embedding
3. Simple search better than a hybrid approach?

2. Rephrased Research Questions

I have reformulated the first question, to cover all concerns with it. I express my reasoning and justification below.

1. **How can multi-modal documents**—specifically product catalogs containing text, images, and tables—be represented in a unified vector space that preserves all embedded information?
2. **Is a single, unified vector embedding more efficient** than separate, multi-vector embeddings for a single document?
3. **Does a simple search mechanism outperform a hybrid approach** when querying this multi-vector vector space?

3. Questions to be answered

From the meeting notes i took i expressed this set of questions that i hope my overview answers

3.1 What makes my approach unique?

How does it differ from existing PDF-to-text frameworks?

In particular, what value do table and image processing add compared to solutions that ignore or only partially handle these modalities?

3.2 What are the benefits of dual-model table processing?

Why use one encoder for structured (grid-style) tables and another for semi-structured (questionnaire-style) tables?

How does this strategy improve fidelity and robustness?

3.3 How does our work fit into existing literature?

Which frameworks or papers already address parts of this problem, and where do they fall short?

In what way does our pipeline go beyond “x,” and why is our “y” better?

3.4 Is our method efficient and worthwhile?

Given the computational overhead of table-to-text conversion, does the added accuracy justify the cost?

How do we measure and demonstrate that trade-off?

4. How can multimodal documents be represented in a unified vector space that preserves all embedded information?

Motivation and Novelty

Most existing PDF-to-text frameworks follow one of two paths: they either extract plain text and ignore tables entirely, or they embed images alongside text in a multimodal space while still neglecting tabular data. Even advanced solutions like Ragflow support image embeddings but omit tables, and recent models focused on structured tables still fail on irregular, real-world layouts. *Look at literature section for more information*

Our framework goes beyond these limitations by:

- **Semantic table handling**: Rather than converting tables into flat text, we translate them into structured, machine-readable representations. These representations feed directly into text decoders, ensuring no tabular information is lost or mangled.
- **Dual-model table processing**: We employ two specialized encoders—one optimized for large, regular grids (e.g., Excel-style tables) and another for semi-structured, questionnaire-like tables with missing values or irregular layouts.
- **Image-to-text conversion**: All images, charts, and graphs are transformed into descriptive text, eliminating the need for multimodal embeddings and simplifying downstream classifiers. This keeps the pipeline flexible and resource-efficient without sacrificing informational fidelity.

In contrast to prior work such as “Struct2Text” approaches that only handle well-aligned tables, our method systematically addresses the chaotic reality of PDF-extracted content. As the authors of [ArXiv:2301.02071] themselves note, “effectively bridging the gap between structured tables and text input remains underexplored.” Our research fills this gap by fully leveraging both structured and unstructured tabular data within a single, end-to-end pipeline.

Practical Considerations

- **Efficiency**: Table-to-text conversion is computationally intensive, and we acknowledge this overhead. However, our benchmarks will demonstrate that the added accuracy and completeness justify the extra cost.
- **Benefit**: By embedding every piece of information—text, images, and both kinds of tables—our framework ensures richer, more accurate representations. This, in turn, boosts the performance of search, summarization, and other downstream tasks. (hopefully)

Sr. No.	Emp.Name	City	Basic Sal	DA	Travelling Exp	Variable
5	Vinod	Banglore	15,474	3,994	7,662	11,702
3	Jhony Mishra	Bhopal	17,402	4,966	6,631	10,996
13	Lalit Modi	Delhi	19,953	3,790	6,018	11,457
14	Rahul	Haryana	14,258	3,429	7,032	11,751

Image 1: Excel-style structure

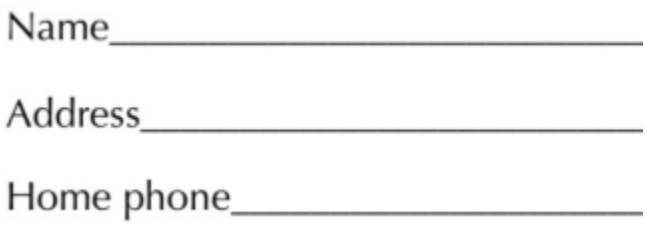

Image 2: questionnaire-like tables

Comparison to Existing Methods and Additional Literature

Most PDF-to-text pipelines either ignore tables or treat only neatly structured grids. Below is a summary of leading tools and their limitations:

- **Generic PDF-to-Text (html-to-pdf.net)**: Extracts plain text but omits tables entirely. The few systems that do support tables rely on highly organized flows, making them unsuitable for real-world, irregular layouts.
- **Ragflow**: Offers multimodal embeddings for images alongside text, yet it completely bypasses tables, leaving a critical modality unaddressed.
- **ArXiv:2408.09869**: Demonstrates advanced text reading capabilities but provides no mechanism for interpreting table structures or content.
- **Unstructured-IO**: Excels at ingesting images and free text, but like most frameworks, it lacks any table processing support.
- **LLaMA Cloud Services parser**: Parses text blocks effectively but fails to recognize or translate tabular data.

These examples highlight the gap in handling tables—readers either skip tables or treat them as text blobs. The proposed framework is the first (that i know of) to fully incorporate both structured and unstructured tables into an end-to-end PDF-to-text pipeline without sacrificing multimodal fidelity.

5. Is a single, unified vector embedding more efficient than separate, multi-vector embeddings for a single document?

The question of whether a single combined embedding outperforms separate vectors for each modality had no objections from Moharram.

To evaluate this, we will build our retrieval system twice: once using a unified vector for each document, and once using a multivector approach. By running identical queries against both configurations, we can compare the accuracy of returned results and quantify any efficiency gains or losses.

6. Does a simple search mechanism outperform a hybrid approach when querying this multi-vector vector space?

Similarly, we will test a basic cosine-distance search against a more sophisticated hybrid algorithm. Both will operate on the same multi-vector index, enabling a direct comparison of precision, recall, and response time. This approach will reveal whether the complexity of a hybrid search is justified by improved retrieval performance.

Remember to check report *[MsTh] Methodology Overview* for more information and additional literature regarding each of the components discussed in this report