Information Extraction From Fiscal Documents Using LLMs

Vikram Aggarwal
viki@google.com
Google Inc
Mountain View, USA

Jay Kulkarni*
jayvilaskulkarni@gmail.com
XKDR Forum
Mumbai, India

Aditi Mascarenhas*
mascarenhasaditi@gmail.com
XKDR Forum
Mumbai, India

Aakriti Narang*
narangaakriti9@gmail.com
XKDR Forum
Mumbai, India

Siddarth Raman*
raman.siddarth@gmail.com
XKDR Forum
Mumbai, India

Ajay Shah*
ajayshah@mayin.org
XKDR Forum
Mumbai, India

Susan Thomas*
sthomas.entp@gmail.com
XKDR Forum
Mumbai, India

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities in text comprehension, but their ability to process complex, hierarchical tabular data remains underexplored. We present a novel approach to extracting structured data from multi-page government fiscal documents using LLM-based techniques. Applied to annual fiscal documents from the State of Karnataka in India (200+ pages), our method achieves high accuracy through a multi-stage pipeline that leverages domain knowledge, sequential context, and algorithmic validation. A large challenge with traditional OCR methods is the inability to verify the accurate extraction of numbers. When applied to fiscal data, the inherent structure of fiscal tables, with totals at each level of the hierarchy, allows for robust internal validation of the extracted data. We use these hierarchical relationships to create multi-level validation checks. We demonstrate that LLMs can read tables and also process document-specific structural hierarchies, offering a scalable process for converting PDF-based fiscal disclosures into research-ready databases. Our implementation shows promise for broader applications across developing country contexts.

Keywords

LLMs, tabular data, information extraction

ACM Reference Format:
Vikram Aggarwal, Jay Kulkarni, Aditi Mascarenhas, Aakriti Narang, Siddarth Raman, Ajay Shah, and Susan Thomas. 2025. Information Extraction From Fiscal Documents Using LLMs. *Being Presented at the AI for Financial Inclusion, Risk Modeling and Resilience in Emerging Markets workshop at ACM ICAIF 2025 Singapore (ACM ICAIF '25).* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM ICAIF '25, Singapore

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 Introduction

While LLMs excel at processing natural language, their ability to process tabular data is poorly understood. A lot of financial disclosures, bills, and fiscal documents continue to be published in PDF formats containing complex tables that are not easily machine-analyzable. While the source information exists in the PDF, data analysis requires human knowledge to understand the document structure and parse it correctly. For Indian fiscal data, traditional OCR methods fail due to inconsistent encoding of both the text and numbers. While LLMs have shown promise in text extraction, their ability to accurately extract structured tabular data is not well researched. One reason for less research into this topic is the lack of availability of large, parallel corpus with PDF documents and parsed, structured text. The lack of a parallel corpus makes it difficult to train models to understand PDF files, and it also makes it difficult to verify the accuracy of information extraction.

In India, government fiscal documents are typically released as lengthy PDFs. These PDF files are a page-by-page materialization of complex hierarchical fiscal tables. While the PDF files contain all the information and are readily available, the page-wise format is not readily queryable, limiting its use in sophisticated, widespread automated analysis. This is true at the Union level, the state level, the municipal level and across public sector firms and parastatals. LLMs have shown great promise at document understanding and summarization but have not been beneficial at wide-scale extraction of information from public governance sources.

We focus on the task of using LLMs for extracting machine-analyzable information from publicly available PDF files of Indian states' finances. We develop an LLM-based process that can accurately extract vital fiscal information for downstream storage and analysis.

This is how our paper tackles the extraction problem:

- We narrow the problem to a single state of interest, and show that the format of fiscal information provided by this state is representative of the complexity of information extraction.
- We outline our process for extracting machine-analyzable information from this state, and list specific improvements in the overall process.

- We demonstrate how to identify the accuracy of the information extracted, even when no ground truth dataset exists. Relying on the structure of fiscal documents, our mechanism also finds the locations where the information extraction failed.

1.1 The Challenge

Fiscal documents disclose government expenditure, but their format renders them effectively unusable for systematic research or public oversight. Traditional OCR approaches fail because:

- Tables span multiple pages; tables have varying structures.
- The files are large: roughly hundreds of pages. Since tables span multiple pages, page-by-page extraction missed vital information.
- Regional languages mix with English in inconsistent patterns.
- Units and formatting vary across documents.

For researchers and civil society organizations aiming to analyze public finances, this creates a significant barrier. Manual data entry is labor-intensive and error-prone. Converting these documents into structured, machine-readable formats through a mix of text and image processing tools is a laborious task that requires manual intervention and verification, as well as domain expertise with public finance. As a result, there are no comprehensive, research-ready databases of government finances in India (or most developing countries) that can be readily analyzed by researchers.

1.2 The Opportunity

Unlike general text extraction, these deep hierarchical fiscal documents contain a natural validation mechanism: columns must sum to reported totals at upper hierarchical levels. This creates a unique opportunity to develop verifiable LLM-based extraction systems where extraction accuracy can be algorithmically validated.

Once structured data is extracted, and given an accuracy score or guarantee, it is then research-ready. Economists, civil society organizations and journalists can use this data to tag different heads of expenditure, and categorize spending by function, geography, department, and other dimensions. This allows for an analysis of government spending patterns, efficiency, and shifts in priorities over time.

Finally, since LLM performance on multi-lingual, hierarchical, multi-page tabular data is known to be poor, this process allows for the creating of a large corpus of parallel PDF and structured-text data that can then be used for pre-training LLMs. Existence of this corpus will drive quality improvements in LLMs on such tabular information extraction and understanding tasks.

2 LLM approaches

LLMs have been used for data extraction and data summarization, and also for Q&A tasks. Early work by Dong et al. [2] covered common formats like web documents, spreadsheets and CSVs, but PDFs were either under represented, or entirely missing. A recent survey on small, English language tables from Wikipedia by Sui et al. [6] finds that LLMs do not have a good understanding of table structure. Currently, most research LLMs focuses on understanding tabular data primarily for the question answering task. Most papers (ex. Zhao et al.[8], Bhandari et al. [1]) focus on machine readable formats like JSON, or text formats delimited by pipe characters. Even working with simpler formats, Zhao et al. [8] demonstrate that tabular data with hierarchical information is poorly understood by current LLMs and the LLM responses are hallucinated. Lu et al. [3] published a survey of LLM techniques showing that academic research has been focused on easier, text-based document types (text, JSON, HTML), and has not fully investigated image, or page-based formats like PDF, and has not fully explored real-world complexity: multi-page tables, multi-lingual tables, hierarchical tables, and PDF files spanning multiple pages. The authors conclude, "Future table LLMs should adapt quickly and cheaply to realworld business needs. Research directions include synthesizing high-quality training data that reflects the diverse needs of specific domains by cost-effective methods." In this paper, we demonstrate a technique for constructing datasets at high accuracy for a highly complex, real-world use-case.

There have been efforts to create evaluation datasets (ex. [4, 7, 10]) with complicated tables representative of real-world use-cases. At its most complex, the TableEval [10] dataset has dual lingual (English and Chinese), hierarchical tables generated from computer spreadsheets provided as JSON or HTML tables. The largest table in that dataset (table 520) has 113 rows and 3 columns, and is a 10 kilobyte spreadsheet. TableEval is intended for Q&A tasks rather than direct information extraction. In contrast, our technique can tackle tables directly from PDF files, each containing hundreds of pages. The PDF files used in our process contain many tables of varying layouts, and tables can spread across pages and contain hundreds of rows. They have interspersed English and Indic characters with inconsistent coding, this is considerably harder compared to previously studied datasets.

3 Methodology

3.1 Key Innovations

Our approach introduces several novel techniques:

We use LLMs to extract tabular data from the fiscal documents. While straightforward prompting can work for simple tables, complex hierarchical tables like fiscal documents require additional techniques to ensure accuracy and consistency. To ensure high-quality extraction in a format that is also algorithmically verifiable, we introduce the following innovations:

(1) **Image-based processing**: Converting PDF pages to high-resolution JPGs (300 DPI) improves LLM comprehension compared to direct PDF input or text-based OCR. Converting a document to image removes all text metadata, which then forces the LLM to perform OCR to recognize text terms, rather than reading the (potentially erroneous) text metadata. However, this makes the problem harder as context from previous pages needs to be carried forward manually.
(2) **Sequential context**: Each page receives the previous page's extracted data as context, enabling state carry-forward across page boundaries. This allows us to parse large PDFs that exceed the context window of LLMs.
(3) **Multi-level validation**: Algorithmic checks verify summation consistency at the various levels of fiscal reporting:

Object Head, Detailed Head, Sub Head, Minor Head, Sub Major Head and Major Head levels.

(4) **Meta-prompting**: Our initial prompt provides domain context about fiscal documents, and provide few-shot examples to the LLM, to get the LLM to write the extraction prompt.
(5) **Intelligent cleaning**: A semantic CSV cleaner uses row-type understanding (Header/Data/Total) to detect and correct column misalignment.

4 Implementation: Karnataka Finances 2020-21

4.1 Problem Statement

Indian fiscal documents contain the following challenges:

(1) **Multi-linguality**: India is a vast, diverse country with 22 official languages written in different scripts. States conduct business in English, in addition to an official state language. Fiscal information is usually published in English, but can also carry the state's official language. In some cases, the source documents are only published in the state's language, and not in English. Further, Indian languages are "low-resource" languages, ie, they suffer from lack of a high quality corpus on the web. This leads to worse LLM performance on both information extraction and generation tasks. Singh et. al [5] compare LLM performance on Indian languages, and demonstrate that many low-resource language exhibit worse performance compared to high-resource languages like English.
(2) **Document Encoding**: Since the documents are published as PDFs, they can carry font information. We find that the regional language is often encoded incorrectly, or is encoded as ASCII and font codebooks are used to display the regional language. Information extraction from such malformed documents is difficult, as machine extraction "sees" ASCII in the metadata. This is true of both metadata-extraction tools (pdf2text) and language models. The only way to avoid this is to convert the document to images, which greatly increases the size of the input, and carries other limitations like overflowing the context window of an LLM.
(3) **Document Size**: One salient characteristic of fiscal data is the vast number of pages in a single document. Documents routinely have more than a hundred pages, and PDF files with more than 500 pages are common. An information extraction technique has to handle this immense size. As mentioned above, a naïve approach involving LLMs chokes on overflowing the context window of current gen LLMs. As an example, table 3 on page 5 shows the number of pages in the PDF of files that we have analyzed.
(4) **Table Structure**: The source documents contain tables of different types. The initial tables contain the top-level hierarchy, and overall totals. Subsequent tables expand on this with department-level detail. Any naive prompt first has to specify this complexity.
(5) **Table Locations**: Fiscal tables often span many pages. Page headers and footers have to be ignored, and the extraction process needs to remember where a table starts and ends. Further, pages can contain multiple tables, as a single page can end a large table from prior pages, fully contain one or more small tables, and start another table that continues for many pages. Any single-page extraction prompt needs to consider this level of complexity. While most pages are in portrait orientation, sometimes wide tables are rotated, and so intervening pages are in landscape orientation.
(6) **Non-standard Number Formatting**: Indian documents use lakhs, crores, thousands, mixed use of commas and periods, and sometimes non-Arabic numerals.
(7) **Scanned Documents**: Some documents are scanned images, others are digitally generated PDFs.
(8) **Hierarchical Tables**: There are multiple nested hierarchies, and not all levels are consistent within or across states.
(9) **Inconsistent Formatting**: Merged cells, inconsistent spacing, additional textual context in the middle of tables that may require semantic understanding to parse correctly.
(10) **Repeated Values**: Similar numbers repeated in different categories (Revenue / Capital, Voted / Charged).

Karnataka's fiscal documents are one of the more complex state documents, exhibiting 8 out of 10 of these challenges. Karanataka's documents are machine generated (Not #7 Scanned Documents), and the numbers are in Arabic numerals (Not #6 Non-standard Number Formatting).

4.2 Approach

Creating the extraction prompt has the following steps, to be followed in order:

(1) **Document structure** We provide the LLM with a background on Indian government fiscal structures, including definitions of Major Head, Minor Head, Object Head, and the typical hierarchical relationships. (To help with the explanation below, this hierarchy relationship is visualized in Figure 1 on page 4.) The LLM is also provided the full PDF document to get the various table types and nesting hierarchies.
(2) **Table structure** We take the document structure, and prompt the LLM to create CSV schemas for each type of table, specifying column names, data types, and expected formats for numerical and categorical data.
(3) **Prompt generation** We give the LLM illustrative pages from the document, with the CSV schemas from Step 2, and ask it to generate an extraction prompt.

This prompt is then used to extract each page of the document sequentially, passing the extracted data from the previous page as context for the current page. For each page, the LLM identifies the table type, applies the relevant schema, and extracts the data into that specific CSV format. If there are multiple table types on a single page, the LLM segments the page accordingly and applies the appropriate schema to each segment.

4.3 Output Schema

For Karnataka, the LLM generates five CSV archetypes:

(1) Sub-Major Head
(2) Minor Head
(3) Sub Head
(4) Detailed Head

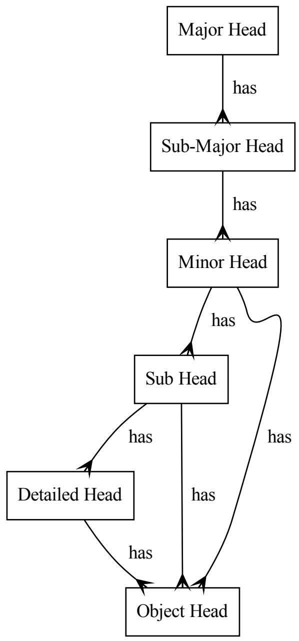

Figure 1: Hierarchy in Karnataka finances

(5) Object Head

4.4 Cleaning

Despite our best efforts to insist on column structure, and specifying the exact schema of output, and getting the results in a structured format, the raw LLM output contains misalignment due to merged cells, inconsistent spacing, and occasional misclassifications of row types. We implement a semantic CSV cleaner that uses the understanding of row types (Header/Data/Total) and row levels to detect and correct column misalignment. This cleaner ensures that each row adheres to the expected schema, correcting for common errors such as shifted columns or misplaced totals.

4.5 Validation Results

Since we do not have ground truth, we perform two types of internal consistency checks as a validation of the extraction process:

- **Numerical Consistency**: By verifying sums of Object Heads against corresponding Detailed Head totals, we ensure within-schema consistency. By matching Minor Head totals from two different schemas we ensure across-schema consistency. It is possible to perform the within- and across- schema numerical consistency checks at all levels of hierarchy.
- **Structural Soundness**: We calculate the Tree Edit Distance Similarity (TEDS), an established metric from Zhong et al. [9] to compare hierarchical structure. TEDS measures how accurately fiscal heads across different schemas match identical structures at the same depth. Here, the numbers are irrelevant, and we're looking to match the hierarchical structure from various sources to validate the extraction process.

Vol.	Validation Type	Checks	Passed	Pass Rate %
1	Object → Detailed Head	374	317	85%
	Two-source Minor Head	154	146	95%
	Overall	528	463	88%
2	Object → Detailed Head	337	284	84%
	Two-source Minor Head	126	118	94%
	Overall	463	402	87%
3	Object → Detailed Head	199	149	75%
	Two-source Minor Head	90	84	93%
	Overall	289	233	81%
4	Object → Detailed Head	189	147	78%
	Two-source Minor Head	60	59	98%
	Overall	249	206	83%
5	Object → Detailed Head	275	203	74%
	Two-source Minor Head	115	113	98%
	Overall	390	316	81%
6	Object → Detailed Head	99	79	80%
	Two-source Minor Head	56	55	98%
	Overall	155	134	86%
7	Object → Detailed Head	161	125	78%
	Two-source Minor Head	76	73	96%
	Overall	237	198	84%
	All Volumes	2311	1952	84%

Table 1: Validation results

File	Pages	Input Tok.	Thought Tok.	Output Tok.
Volume 1	227	1567223	1898941	443077
Volume 2	179	1412790	1486710	430499
Volume 3	139	977091	1140116	299388
Volume 4	142	1052354	1207653	317168
Volume 5	181	1452697	1499687	421164
Volume 6	74	466169	583537	123694
Volume 7	112	842688	1010741	235464

Table 2: Gemini 2.5 Pro Token count

Table 1 presents the results for **numerical consistency** checks for all seven volumes of the Karnataka state finances in Year 2020–2021. Due to the multiple independent sources of numerical validation, errors in consistency checks can be traced back to specific pages and rows, allowing for targeted manual review or re-extraction. Our pipeline prints the locations where the specific failures occurred. As an example, here are the locations of failures from the Object → Detailed report on Volume 1:

```
Major_Head          2039
Description         Total 09
Page                3
Status              FAIL
```

```
Accounts_2018_19_Match PASS
Budget_2019_20_Match   FAIL
Revised_2019_20_Match  FAIL
Budget_2020_21_Match   PASS
```

Table 3 presents results for our implementation of **structure soundness** through the Tree Edit Distance-based Similarity (TEDS) measure from the paper by Zhong et al. [9]. For each of the five extracted schemas within a volume, tree structures were created for each Major Head hierarchy. The tree structures for two schemas at the same depth were then compared. Since the two schemas are getting generated from different pages in the PDF document, we effectively cross-check fiscal-head hierarchies from two different locations in the same source. A TEDS score of zero implies that the structures are identical. We divide the number of instances with zero TEDS against all calculated TEDS for each volume and report that as the single accuracy number in table 3. For all the volumes of the Karnataka government, we have been able to achieve an accuracy of 73% to 96%.

File	Pages	Accuracy
Volume 1	227	95.24%
Volume 2	179	73.68%
Volume 3	139	88.00%
Volume 4	142	83.33%
Volume 5	181	96.77%
Volume 6	74	91.30%
Volume 7	112	79.17%

Table 3: Tree Edit Distance Similarity (TEDS)

Let's summarize the key advances made through this validation:

(1) The ability to get accuracy metrics in the absence of ground truth data.
(2) The ability to identify locations where the extracted data does not match the original PDF. Errors at these locations can then be corrected with hand-labeling. Where the errors are small or localized, the extracted data can be used if failures occur in locations that are not under analysis.
(3) The understanding of multi-level hierarchy during extraction, and corresponding validation of the hierarchy through different sources in the same PDF document.

5 Future Directions

We have demonstrated a technique for extracting information from state fiscal documents by using a single Indian state (Karnataka) as an example. Our research goal is to extend this work across all Indian states and over many years of fiscal disclosures, to create a robust, analyzable dataset of state finances. To achieve this goal, here are the future directions for our work:

- **Automated meta-prompting**: Develop a pipeline where the LLM:
  - Analyzes document structure (table types, page ranges).
  - Generates CSV schemas per table type.
  - Creates extraction prompts automatically.
  - Cleans up the output using a mix of rule-based and LLM-based cleaning.
  - Validates and iterates until the validation passes.
- **Cross-state robustness**: Test across multiple Indian states with varying document structures.
- **Multi-year consistency**: Ensure format stability across fiscal years.
- **International applicability**: Extend to fiscal documents from other developing economies.

5.1 Application Areas

While this paper has a narrow focus: the task of extracting information from state finances; Our overarching goal is to use this data to understand and improve the functioning of states. This work will find value in:

- Public procurement analysis
- Budgeting efficiency studies (Budgeted vs Revised vs Actuals)
- Expenditure priority tracking over time
- In-year fund reallocation analysis
- Functional/departmental/geographic classification

5.2 Reproducible Research

All materials, including the fiscal documents, source code, validation scripts, extracted information and instructions are available under MIT license at: https://www.github.com/xkdr/acm_icaif_2025_paper.

6 Conclusion

We have demonstrated that LLMs can successfully read and understand complex hierarchical tables when provided with appropriate domain knowledge, sequential context, and validation mechanisms. Our approach understands the structure of 200+ page PDF documents, and extracts it into research-ready machine-readable files with a numerical accuracy of 84%, and structural accuracy ranging from 73% to 96%. Our mechanism also provides the locations of data extraction failures, creating a technique that can be used to generate verifiable seed datasets for downstream analysis.

The key insight is that LLMs bring semantic understanding to table extraction – they don't just perform pattern matching but leverage domain knowledge acquired from large-scale pretraining. Combined with built-in validation through summation checks, this enables reliable automation of a task previously requiring extensive manual effort.

While it has been applied here to Indian state finances, the methodology is broadly applicable where complex tabular data is published in non-machine-readable formats and where internal consistency checks can be defined. This opens up new avenues for leveraging LLMs in public finance research, transparency initiatives, and data democratization efforts, but also in generic table extraction tasks across domains.

Acknowledgments

We thank Yi-fan Chen and Dima Kuzmin for discussions and support. We thank Atibhi Sharma and Meghna Saxena for their contribution to this project.

References

[1] Kushal Raj Bhandari, Sixue Xing, Soham Dan, and Jianxi Gao. 2024. On the Robustness of Language Models for Tabular Question Answering. *CoRR* abs/2406.12719 (2024). https://doi.org/10.48550/arXiv.2406.12719

[2] Haoyu Dong, Zhoujun Cheng, Xinyi He, Mengyu Zhou, Anda Zhou, Fan Zhou, Ao Liu, Shi Han, and Dongmei Zhang. 2022. Table Pre-training: A Survey on Model Architectures, Pre-training Objectives, and Downstream Tasks. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, Lud De Raedt (Ed.). International Joint Conferences on Artificial Intelligence Organization, Singapore, 5426–5435. doi:10.24963/ijcai.2022/761 Survey Track.

[3] Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. 2025. Large language model for table processing: a survey. *Frontiers of Computer Science* 19, 2 (12 Jan 2025), 192350. doi:10.1007/s11704-024-40763-6

[4] Reducto. 2024. https://reducto.ai/blog/rd-tablebench

[5] Harman Singh, Nitish Gupta, Shikhar Bharadwaj, Dinesh Tewari, and Partha Talukdar. 2024. IndicGenBench: A Multilingual Benchmark to Evaluate Generation Capabilities of LLMs on Indic Languages. In *Annual Meeting of the Association for Computational Linguistics*. ACM, Singapore. https://api.semanticscholar.org/CorpusID:269362363

[6] Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study. arXiv:2305.13062 [cs.CL] https://arxiv.org/abs/2305.13062

[7] Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, Tongliang Li, Zhoujun Li, and Guanglin Niu. 2025. TableBench: a comprehensive and complex benchmark for table question answering. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence (AAAI'25/IAAI'25/EAAI'25)*. AAAI Press, Singapore, Article 2841, 10 pages. doi:10.1609/aaai.v39i24.34739

[8] Bowen Zhao, Changkai Ji, Yuejie Zhang, Wen He, Yingwen Wang, Qing Wang, Rui Feng, and Xiaobo Zhang. 2023. Large Language Models are Complex Table Parsers. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali (Eds.). Association for Computational Linguistics, Singapore, 14786–14802. doi:10.18653/v1/2023.emnlp-main.914

[9] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno-Yepes. 2019. Image-based table recognition: data, model, and evaluation. *CoRR* abs/1911.10683 (2019). arXiv:1911.10683 http://arxiv.org/abs/1911.10683

[10] Junnan Zhu, Jingyi Wang, Bohan Yu, Xiaoyu Wu, Junbo Li, Lei Wang, and Nan Xu. 2025. TableEval: A Real-World Benchmark for Complex, Multilingual, and Multi-Structured Table Question Answering. arXiv:2506.03949 [cs.CL] https://arxiv.org/abs/2506.03949