Extracting Financial Data From Unstructured Sources: Leveraging Large Language Models

Huaxia Li
Department of Accounting and Information Systems
Rutgers, the State University of New Jersey
huaxia.li@rutgers.edu

Haoyun (Harry) Gao
Department of Accounting and Information Systems
Rutgers, the State University of New Jersey
hg402@rutgers.edu

Chengzhang Wu
School of Business
Stockton University
chengzhang.wu@stockton.edu

Miklos Vasarhelyi
Department of Accounting and Information Systems
Rutgers, the State University of New Jersey
miklosv@business.rutgers.edu

September 2023

JEL Classifications: M41; O31; C81

Keywords: Data extraction; LLM; GPT-4; Governmental Reports; Design Science
Extracting Financial Data From Unstructured Sources: Leveraging Large Language Models

ABSTRACT

This research addresses the challenge of extracting financial data from unstructured sources, a persistent issue for accounting researchers, investors, and regulators. Leveraging large language models (LLMs), this study develops a framework for automated financial data extraction from PDF-formatted files. Following the design science methodology, this research develops the framework through a series of text mining and prompt engineering techniques and further applies it to governmental annual reports in PDF format. Pilot test results indicate that the framework achieves a 100% accuracy rate within a short period of time when extracting key financial indicators. This study contributes to the evolving literature on applying LLMs in accounting and finance, while also providing a practical tool for both academic and industry applications.

I. INTRODUCTION

Accounting is a study built on the quantitative expressions of economic phenomena (Davidson 1966). Therefore, data has played a significant role in facilitating the understanding, auditing, and prediction of business activities. The emergence of machine-readable datasets, such as Compustat in 1962, has transformed the accessibility of accounting data for all stakeholders, thereby also fueling research in accounting and capital markets (Teoh 2018). Despite the availability of existing datasets, there are still numerous areas where financial numbers are stored in unstructured sources. These include, but are not limited to, financial data presented within Portable Document Format (PDF)-formatted governmental annual comprehensive financial reports (ACFRs) (Li et al. 2023), quantitative information negotiated within contracts (Yan and Moffitt 2019), or greenhouse gas emission volumes disclosed in corporate social responsibility (CSR) reports (Jiang, Gu, and Dai 2023). Advancements in large language models (LLMs) at the end of 2022 offer great potential to overcome existing limitations by transforming human-
generated unformatted information into machine-readable standardized databases (Gu et al. 2023). This study takes the first step in developing an LLM-enabled framework that can extract financial data from unstructured sources, thereby providing valuable insights for market participants, policymakers, and researchers.

Specifically, we develop a framework enabled by LLMs that can process PDF-formatted data and extract predefined financial data from it. To provide systematic guidance, our framework incorporates data preparation, prompt engineering, batch querying, and database construction. We further demonstrate the framework's practical application by applying it to extract key financial indicators from the local government's ACFR and compare the accuracy and efficiency with manual processes. To complement the evaluation of our framework, we collaborate with the Government Finance Officers Association (GFOA) to obtain expert reviews on the design and implementation of the framework.

This study follows the six-step design science research methodology proposed by Peffer et al., (2007). First, we acknowledge the existing limitations of unstructured data in providing financial information and recognize the opportunity to design an LLM-enabled framework to address this issue (Identify Problem and Motivation). Next, we define the objectives for solving the problem, including validating the effectiveness of the artifact and enhancing its efficiency in data extraction (Define Objectives of a Solution). We then develop and describe the artifact (Design and Development) and test its application on a real-world data extraction task involving ACFRs (Demonstration). To evaluate and refine the framework, we compare the extracted results with the actual values in the documents and also consult with the GFOA for expert advice (Evaluation). Finally, we document both the artifact and its application in this manuscript (Communication).
The framework leverages LLMs, emerging technologies capable of understanding human language, to identify account names within financial statements and extract their corresponding values. To maximize the LLM’s capabilities, we follow methods developed in the field of Natural Language Processing (NLP) to fine-tune the model before integrating it into the framework. We also incorporate optimal prompts into functions that can be adapted to various LLM models and extraction tasks, facilitating batch processing of a large number of files with an extensive range of data points.

We demonstrate the framework by extracting 19 items from each of 8 local county ACFRs in California, which results in a total of 152 items. The first set of test results reveals an accuracy rate of 96.1% when compared with the actual values from the reports. Upon manual examination, we notice that all the misidentifications are concentrated on two specific items. Therefore, we refine the relevant prompts by providing clearer instructions and examples to guide more accurate extraction. The second test yields a 100% accuracy rate.

To provide further evaluation of the framework, we seek expert opinions from the GFOA. Specifically, we hold regular online meetings with GFOA representatives, during which we present the framework, share preliminary results, and demonstrate the process to accountants, technologists, and managers. After receiving feedback, our research team internally discusses revisions to the framework. The experts unanimously agree that the framework is effective and efficient in extracting financial data from unstructured sources.

This research contributes to the literature in several ways. First, we develop and test a framework that leverages LLMs to extract financial data from unstructured sources. Researchers from multiple disciplines can apply our framework to obtain new data, either to expand their current research pipelines or to explore new research areas.
Second, the proposed framework contributes to the body of research that examines the effectiveness of emerging technologies in accounting and auditing (Dowling and Leech 2014; Hodge, Mendoza, and Sinha 2021; Issa 2018; Sun and Vasarhelyi 2018; Wu and Dull 2021; Yan and Moffitt 2019). The findings of this study demonstrate that technology-enabled methods can enhance accounting research by improving efficiency and reducing data collection costs.

Third, the proposed method echoes the call for accounting research that is relevant to practice. The accounting literature has long been criticized for being disconnected from practical concerns (Burton et al. 2022; McCarthy 2012; Rajgopal 2021; Waymire 2012). Our framework serves as a solution to address one of the practical problems faced by stakeholders, thereby increasing its relevance to practice.

The remainder of this paper is organized as follows: Section 2 introduces the background of this research and summarizes existing literature. Section 3 outlines the motivations for developing the artifact, while Section 4 further sets the objectives of the proposed framework. The artifact is developed in Section 5 and is subsequently illustrated in Section 6. Section 7 presents the evaluation results, and Section 8 provides discussion. Finally, Section 9 concludes the paper.

II. BACKGROUND

Large Language Models

The field of artificial intelligence (AI) has long sought to develop machines capable of reasoning and thought. Different attempts have been made to understand human natural language in the NLP domain, LLMs being one of the latest and the most phenomenal. The first LLM was the "Transformer" model, introduced in the paper "Attention Is All You Need" (Vaswani et al. 2017). LLMs are developed both to enhance context comprehension and to generate new content.
Recent breakthroughs in LLMs, such as the Chat Generative Pre-trained Transformer (ChatGPT) launched by OpenAI in November 2022 (OpenAI 2022), have generated significant attention and sparked discussions about the potential of LLM-powered AI tools to complement or augment human workers in various domains. It is important to note that LLMs are not limited to question-answering, exam-taking, or creative storytelling. For instance, OpenAI’s latest foundation model, Generative Pre-trained Transformer 4 (GPT-4), is capable of handling diverse tasks, including the analysis of visual inputs (Liu et al. 2023).

The development of a foundation LLM such as GPT-4 involves a process known as model pre-training. This is a computationally expensive process that requires extensive training on powerful graphic processing units (GPUs). The model is typically trained on a massive dataset to learn the underlying structures and nuances of the language. Furthermore, the training process may involve distributing the workload across multiple GPUs in a parallel fashion to accelerate the process and manage computational resources more efficiently. These are also very intricate steps and require advanced technical skills. A pre-trained model can be used off the shelf. It is just when you type a question (a “prompt”) to the ChatGPT dialog interface. Such a method of using the model, where it generates responses to prompts it has never seen before without requiring any task-specific training, is known as zero-shot learning.

LLMs may not have been trained in a similar task as what the users want them to perform, like translating from a language that the model has never been trained on. That is where model fine-tuning will be necessary. If the user provides one/a few example(s) to the model before using it, it is called one/few-shot learning. Prompts can be used to fine-tune a model using natural
language. One needs to carefully design each prompt to make sure the model can follow what it is expected to achieve. Such design techniques and principles are called prompt engineering. The concepts discussed above should suffice for the below discussions.

**Literature Review**

LLMs have been applied to understand financial information from textual disclosure data. Sentiment analysis is a topic that has been around since the early days of NLP methods, such as dictionary-based methods where words are categorized into positive and negative ones and then counted. Going beyond mere lexical units counting, Huang et al. (2023) leveraged techniques from the Bidirectional Encoder Representations from Transformers (BERT) model, one of the precedents of GPT, to develop a new LLM called FinBERT that can identify sentiments and topics of financial texts better than traditional, general-purpose tools. Advanced models like GPT-3 or GPT-4 offer even further improvements in sentiment analysis capabilities and can even be used to generate stock return predictions based on news headlines (Lopez-Lira and Tang 2023).

Textual information extraction in the governmental accounting area has been explored using GPT-4, which can extract pension plan information from ACFRs (Gu et al. 2023). Numerical information extraction is different from textual information extraction. While both tasks require contextual understanding, accuracy is paramount when dealing with numbers. However, there is a relative lack of research on using LLMs in the accounting literature. Previous research has either focused on the contextual information surrounding the numerical data (A. G. Kim and Nikolaev 2023), or simply ignored the numerical information entirely (Küster, Steindl, and Goettsche 2023). In the computational linguistics literature, efforts to enable LLMs to capture numerical information
have resulted in new methods that demonstrate improvements in predicting stock price movement (Yang et al. 2022).

In the corporate accounting domain, obtaining accurate financial information remains challenging, despite the availability of well-established databases such as EDGAR, Compustat, and CRSP, which have been the primary data sources for the past half-century. In the early days of Compustat, data accuracy was challenged by researchers, sparking widespread discussions about its implications for research credibility (Rosenberg and Houglet 1974; Chychyla and Kogan 2015; Boritz and No 2020). In more recent years, as computerized data sources become mainstream, new solutions such as adding the eXtensible Business Reporting Language (XBRL) tags have been introduced to improve data quality. These solutions have bolstered investor confidence in financial data. However, they all necessitate a mandate for the adoption and development of industry-specific taxonomies. The challenges are even more significant in less-regulated domains, such as governmental accounting and ESG reporting in the US. While a unified reporting standard would be the most effective solution in the long run, its implementation would be costly and time-consuming and could initially lead to confusion.

The generative capability of GPT models offers unique advantages. For instance, Kim et al. (2023) created a new measure “Bloat”, calculated as the length of a summary provided by ChatGPT (GPT-3.5) relative to its original text. This serves as a proxy for measuring the level of redundancy or lack of informativeness in the original text. De Kok (2023) introduced a five-step framework to conduct textual analysis research in accounting, using the latest GPT-4 model for demonstration. These examples underscore the growing significance of generative language models in advancing research and applications across various fields.
III. PROBLEM IDENTIFICATION AND MOTIVATION

Unstructured data is prevalent in the accounting domain. One of the dominant data formats is PDF documents featuring tables, graphs, numbers, and texts. Our focus is on extracting information from tables within PDF documents with irregular formatting and inconsistent line-item descriptions.

Ultimately, our goal is to provide a general framework for extracting financial data from unstructured sources, especially when a well-developed taxonomy is lacking. We believe that the recent advancements in LLM can make a quantum leap in financial data understanding. We intend to extend our research to address the existing gaps in data availability across various domains using this framework. For market participants, the ability to systematically process and analyze data is crucial for making informed decisions.

IV. OBJECTIVES OF THE ARTIFACT

This artifact is designed to extract information from unstructured financial data. Previous attempts have been made to use tags under the XBRL framework. However, for newer or less-regulated sources, loading PDFs directly would be the only possible solution. This is also an uncharted task, with no existing benchmarks for guidance. To tackle the challenge of inconsistent formatting across different PDF documents, we opt to apply the newly developed LLM tool to achieve both effectiveness and efficiency. For effectiveness, we notice that the lack of consistency in the financial terms poses challenges for accurate information identification. We aim for the extracted data to be matched with the manual extracting results by human experts. At the same time, our goal is to maximize the efficiency of the extraction. To this end, we plan to enable batch processing for the extraction and to develop a function that can be easily used.
The unstructured data we have on hand is the ACFR issued by the U.S. local governments to calculate the risk factors for GFOA; however, there lacks a good database after the year 2014 to achieve this goal (W. J. Kim, Plumlee, and Stubben 2022). Also, even if the tables are all stored in computer-readable format, the description for each line item can vary a lot for the same item. This can be a huge challenge considering that XBRL tags and the taxonomy for the items are absent in the governmental accounting domain in the US.

V. DESIGN THE ARTIFACT

As illustrated in Figure 1, this framework involves the preparation of the source data, the design of prompts, batch querying using the LLM model, and the development of a database to store the extracted data points. The accuracy of the extraction is fundamentally dependent on all of these steps. Therefore, we provide a detailed description of each step to ensure a comprehensive understanding of the framework.

Data Preparation

The first step involves converting all the files from their original formats (.pdf) to machine-readable plain text (.txt) files. This transformation is essential not only for ensuring machine readability but also for facilitating seamless integration by the subsequent data preprocessing and batch querying. To facilitate this format transformation, one can leverage softwares\(^1\), Python packages\(^2\), or online web-based tools\(^3\), depending on the source documents and specific tasks. The choice of method should be justified through a thorough comparative assessment of multiple

\(^{1}\) Software such as Adobe Acrobat, PDFElements, Able2Extract Pro, etc.
\(^{2}\) Python packages such as PyPDF2, Tika, PDFminer, etc.
\(^{3}\) Web-based tools such as CloudConvert, Smallpdf, pdf2go, etc.
solutions. Ideally, the selected method should generate consistent conversion performances than alternative ways.

After acquiring the plain text files, we introduce these documents to LLM for the purpose of Table of Contents (TOC) understanding, which involves segmenting page numbers for distinct content sections. Utilizing LLM's natural language understanding capabilities, we effectively align each section in the TOC with its corresponding page numbers. For content sections that have the potential to stretch across several pages, an additional step is incorporated: LLM is instructed to identify the page number of the immediately succeeding section in the TOC. By procuring this subsequent page number, we can accurately infer the entire page range for the current content section. The specific methods for designing the prompts for LLM will be discussed in the following section.

Sometimes the page ranges parsed from the TOC contain an extensive number of pages because the detailed segmentations are not provided by the original document. Therefore, we further refine those page ranges to one or two pages that are most pertinent to the target data points. This refinement serves to minimize the noise data during extraction, resulting in more accurate outputs. To achieve this, one can manually identify task-specific rules around the target data points and employ Python’s regular expressions\(^4\) to eliminate the irrelevant pages within the range. This approach is also effective on files that lack a TOC; in such cases, the entire file can be treated as a page range for applying this method.

\(^4\) Python regular expression is a specialized syntax used for pattern matching within text. One can use it here to match the rules (keywords) around the target data points.
When dealing with extensively long documents, such as annual reports or CSR reports, one effective approach is to structure the entire document in a dictionary format. This method ensures a well-defined mapping between page numbers and the corresponding page content, thus simplifying the process of extracting pertinent page content based on the TOC understanding results. For instance, a target page identified with the page number $N$ from the TOC understanding can be directly retrieved from the dictionary by searching for the “Key” of $N$; the corresponding “Value” will be the page content.$^5$

**Prompt Engineering**

Prompts are natural language instructions or queries that humans use to interact with LLMs, guiding their responses to achieve specific outcomes (Reynolds and McDonell 2021). The reliability of the response depends on prompt engineering, which denotes the systematic development and optimization of prompts to enhance interactions in alignment with specific objectives or requirements (Gu et al. 2023; Reynolds and McDonell 2021). A key feature of prompt engineering is its ability to guide LLMs using precise instructions, context, and examples (Chung et al. 2022; Küster, Steindl, and Goettsche 2023; Min et al. 2022).

Within our framework, we employ instruction learning, zero/few-shot learning, and Chain-of-Thought (CoT) prompting to enhance the performance of the extraction. In the following sections, each of these techniques will be introduced and explained in detail.

---

$^5$ In dictionary format, the “Key” is the identifier of the data, and the “Value” is the data. Under the context of this method, the “Key” is the page number and “Value” is the page content.
Instruction Learning

Extracting data from unstructured sources requires clear and consistent instructions to prevent misunderstandings or omissions (Li et al. 2023). To achieve this, we first apply instruction learning to fine-tune the LLM model. Instruction learning is a method that is designed to enhance the performance and generalization capabilities of LLMs by fine-tuning them based on tasks described through explicit instructions (Chung et al. 2022; Gu et al. 2023). The goal is to better equip the model to understand and follow a wide range of natural language instructions, enabling it to efficiently tackle and complete real-world tasks, especially when dealing with foundation models that have not been trained for specific tasks.

To implement instruction learning, we design the prompts by categorizing them into three sets of inputs:

**Role and Context**: This is a statement that clearly delineates the role of the LLM in the process and describes the general task. By contextualizing the LLM, it can narrow its focus to a specific knowledge domain and reasoning logic, thus enhancing response accuracy and consistency.

**Rule**: This set of prompts outlines requirements that the LLM must adhere to throughout the task. Requirements that apply to the entire extraction process are formatted and incorporated into this set of prompts. For example, methods designated for identifying target information, alternative strategies for handling exceptions, and the desired output format of the extracted data all fall under this category.

**Task**: The third set of prompts contains detailed descriptions of each extraction item. It's essential to specify the particular items needing extraction from the unstructured source. The more precise the description, the higher the accuracy of the extracted results. For instance, one task could
be phrased as, “What is the total long-term liability of the governmental activities (expressed in thousands) for the city of New York in the year 2020?” Since the foundation model is already trained to understand human languages, the wording of the prompt can be slightly modified in terms of phrasing the task. However, it's crucial to retain a level of detail to ensure accurate extraction.

**Zero/Few-shot Learning**

Data extraction tasks often involve numerous data points and repetitive processing of unstructured data. Consequently, cost becomes a significant consideration when developing models. A time-saving and cost-effective approach should be prioritized to ensure the sustainability of the model. To address this, we employ both zero-shot learning (ZSL) and few-shot learning (FSL) depending on the complexity of tasks.

ZSL highlights the foundation model’s capability to tackle tasks that haven’t been encountered during its training phase. This method allows LLMs to handle new and unseen data by drawing from extensive pre-existing knowledge, without needing explicit training on specific tasks (Kojima et al. 2022). This capability eliminates the often burdensome and expensive requirement for human labeling, which not only reduces expenses but also enhances the scalability of the extraction process. Since LLMs are trained on vast datasets, they possess a deep understanding of worldly knowledge. Combined with ZSL, LLMs can address tasks using contextual cues. Hence, ZSL is ideally suited for tasks that are straightforward and don’t necessitate complex analytical processing. Utilizing this method, we design the prompt by primarily following the instruction learning logic to clearly outline the requirement for each task in the “Task” prompts.

However, certain extraction tasks are not straightforward, such as those requiring the identification of multiple sub-items to derive a total value. These tasks necessitate additional
guidance for accurate extraction. This is where FSL steps in. A modern development in deep learning and NLP, FSL involves fine-tuning models using a minimal number of labeled data (examples) (Brown et al. 2020; Zhao et al. 2021). This approach aids the model in understanding and inferring new instances based on examples, offering a more intuitive learning method. FSL is particularly effective in unique extraction scenarios where training data is sparse. When implementing this approach, rather than merely include descriptions in “Task”, one should integrate several examples along with the descriptions to facilitate accurate extraction.

**Chain-of-Thought Prompting**

More complex extraction tasks often necessitate multiple steps to accurately pinpoint data locations and then extract the data in the desired format. Directly applying instructions without meticulously crafted logic might lead to omissions. To address this, we utilize CoT prompting (Wei et al. 2022), a recent advancement in prompt engineering that is specifically designed to enhance the reasoning capabilities of LLMs.

Particularly, CoT prompting guides a model by providing a series of short, interrelated statements or sentences, serving to direct the reasoning process of the LLM in a manner similar to how a human might approach a task (Gao et al. 2023; Gu et al. 2023). This step-by-step breakdown progresses from the task's inception to its eventual solution, illustrating the model's thought process. CoT prompting has consistently demonstrated improved model performance, especially in tasks demanding detailed reasoning. Moreover, the transparency and interpretability provided by CoT prompting offer users insight into the model's cognitive processes, rendering its decisions more understandable and trustworthy.

To deploy CoT prompting, one should start by emulating human problem-solving approaches, breaking the task into smaller, sequential statements. Since the output of one statement
might serve as the input for the next, it's essential to define relevant variables to store and reference these outputs later. With this logic in place, one can craft customized prompts to generate a chain of logically interconnected reasoning statements that lead to the solution. Additionally, the FSL can be integrated into the process to supply some examples.

**Batch Querying with LLM**

Typical methods in the accounting and auditing literature that apply LLMs usually rely on the user interface (UI) for interaction (Emett et al. 2023; Eulerich and Wood 2023; Föhr et al. 2023; Gu et al. 2023). The benefits of this approach are evident, including high interpretability, easy demonstration, and a no-code environment. However, data extraction tasks often involve analyzing a large number of target data points sequentially within a file. Thus, manually querying each prompt through UI is not cost-effective, particularly when dealing with massive documents. To streamline this process, we integrate optimal prompts into a function that can be readily applied using LLM's Application Programming Interface (API).

Specifically, we follow the aforementioned prompt engineering approaches to design and fine-tune the three sets of prompts, namely “Role and context”, “Rule” and “Task”. We then formalize them into a Python function with parameters including API Key, model, data course, and output location. After running the process, the function automatically iterates through each file in the data source and executes the defined data extraction tasks in the prompts.

This transformation offers several benefits. First, compared to the prevalent method in the literature that involves direct interaction with the LLM via UI (Eulerich and Wood 2023; Föhr et al. 2023; Gu et al. 2023), our function facilitates ease of deployment. Stakeholders, irrespective of their programming expertise, can easily engage with it by supplying the API key and data source and then clicking “Start.” They can then wait to receive the extracted data in the default format.

Electronic copy available at: https://ssrn.com/abstract=4567607
Second, this form of integration simplifies maintenance in the future. In case of errors or updates, the function can be easily traced, enabling quick debugging and enhancement. Unlike a series of prompts intertwined with other processes, this function stands apart from data preparation and database creation. Consequently, modifications to the function won't disturb context settings, allowing for more flexible maintenance. Additionally, with minor adjustments, this function can be adapted to extract other types of data from various sources, such as hazardous emission quantities within the CSR report.

Furthermore, the function enhances the adaptability of our framework to various LLMs. Although developed and tested on only one of the leading LLMs, our framework’s generalizability is not confined to it. Integrating prompts within a function format allows its application across different LLMs. This approach facilitates performance comparisons among various models, enabling users to determine the specifications that best fit their task.

**Database Construction**

Upon completing the batch querying, the extracted data will be consolidated within a Database Management System (DBMS) to enable long-term storage and effortless querying. Prior to constructing the database, it is necessary to preprocess the data extracted from LLM. For instance, original PDF files often represent numerical values in thousands to save space. Consequently, the extracted numerical figures should be aligned to the same unit to facilitate a standardized database. Additional steps for data preprocessing can be determined based on the specific data extracted.
The choice of DBMS should be determined by the data type, data volume, ease of use, and cost. For the purposes of this research, we have employed PostgreSQL as an example DBMS due to its flexibility, stringent adherence to SQL standards, and its status as a free and open-source platform. In PostgreSQL, each record is uniquely identified by a specialized attribute known as a unique identifier (primary key), such as the "GVKEY" (Global Company Key) within the Compustat database. The primary key ensures the consistency of the records among various distributed tables.

VI. ILLUSTRATION ON GOVERNMENTAL REPORTS

This section illustrates an application of our method. Following the steps outlined in section 5, we showcase how to apply the framework to systematically extract financial data from the ACFR of the U.S. local governments.

ACFRs prepared by individual governments are the primary source of comprehensive financial information for U.S. local governments. Other than the ACFR, there is no centralized and publicly accessible database of governmental financial data (W. J. Kim, Plumlee, and Stubben 2022). As a result, extracting key financial data from the ACFR to create a publicly available database is a significant contribution to the literature in accounting, finance, and public policy, as well as for practitioners. To demonstrate this, we leverage our framework to extract key financial data from ACFRs and further assess its extraction effectiveness and efficiency. To further evaluate and improve our framework, we regularly meet with the experts from the GFOA to discuss and modify the framework.
We initiated the illustration by consulting the experts from GFOA to come up with a list of key financial data to be extracted. Specifically, we first relied on the ratios from the traditional ten-point method in assessing municipal fiscal health to select the testing financial data points (Maher 2013). In cases where any items are not disclosed in ACFR under the ten-point structure, we introduced the items from Moody’s schema that evaluate local governments’ health as complementing indicators. The final list of financial data containing 19 items in assessing the local government’s financial situation is provided in Appendix 1. All of them were evaluated and confirmed by the accountants and experts from GFOA. Finally, we randomly selected eight local counties from the state of California to implement the extraction process.

**Data Preparation**

All the ACFRs submitted by US local governments are in PDF format, so the first step is data conversion. We used the software "Able2Extract Pro" to batch convert all the PDF files into plain text format. Since the software integrates Optical Character Recognition (OCR) and other basic page manipulation capabilities, it can automatically recognize information from the scanned pages and restore the pages with a 90-degree rotated layout.

Next, we programmed the Python functions to automatically grab each ACFR’s TOC section into the LLM model to perform the TOC understanding. In this illustration, we apply one of the leading LLMs at the time when this demonstration was conducted, namely GPT-4, to read and understand the TOC. Specifically, our observations indicated that the TOC section within ACFRs usually does not occupy more than the first 165 lines of the converted document. Therefore, the program will input the first 200 lines of each ACFR into GPT-4 for TOC understanding. Since the statements in the ACFR usually occupy more than one page, it is crucial to follow the framework to ensure the LLM can accurately determine the complete range of pages.
for each statement. As illustrated in Figure 2, we first prompt GPT-4 to identify the starting page of the "Statement of Net Position" for the County of Orange, which is Page 41. Next, we further prompt GPT-4 to locate the beginning of the immediate next distinct statement, which starts on Page 43. Finally, we can logically deduce that the "Statement of Net Position" encompasses both Pages 41 and 42.6

It's also unavoidable that certain sections occupy an extended range of pages, such as the “Notes to Basic Financial Statement”, which is typically presented in more than 80 pages. Therefore we further perform the page range refinement by screening the keywords that appeared around the target data points. For example, the extraction of the "Long-Term Obligations for Governmental Activities" from ACFR presents such a challenge, predominantly due to its arbitrary localization within the "Notes to Basic Financial Statement" (Notes) section. Directly utilizing such extensive content as an input to GPT-4 introduces considerable temporal inefficiencies and may also compromise the precision of data retrieval. In response to this challenge, we employed Python regular expressions for refinement. Specifically, we find the pages in the “Notes” part that meets the following three criteria, irrespective of the case sensitivity of the words: (1) The content must contain one of the following keywords: "long-term liabilities", "long-term debts", "long-term obligations", "long-term liability", "long-term debt", "long-term obligation", "summary of long-term debts"; (2) The content has the keyword "additions"; (3) The page must include more than 30 digits. The refinement resulted in the identification of the "Long-Term Obligations for Governmental Activities" from only one or two pages of the “Notes” section.

6 The actual prompt used here in the TOC understanding is illustrated in the next Prompt Engineering section.
Finally, as the ACFRs usually span over 100 pages, in order to fetch information more easily using the page range information from the TOC, we follow the framework to transform the file into a page dictionary using a set of Python codes (mainly regular expressions). With this design, users can limit the LLM input to only the section with the information content. This approach not only reduces LLM operation costs by limiting the input size but also improves extraction accuracy by focusing on the most relevant content. We began by splitting the document into plain text chunks, using empty lines as separators. Since page numbers were usually at the top or bottom of each page, we checked the first and last lines of each chunk for these numbers. We skipped MD&A sections as they could have had separate numbering. If we encountered formatting issues that prevented the identification of page numbers, such as years or financials from tables instead of page numbers, we automatically incremented the page number for that chunk by one. If no numbers could be extracted, the page number remained the same.

Page numbers can identify all the chunks that are assigned with certain page numbers, and once we have a range of page numbers for a certain section, all chunks can be stitched together to form a string that will be fed into later steps.

Prompt Engineering

With the relevant mapping established between the page number and its content, we further utilize GPT-4 as an illustrative LLM to extract the exact figures for each item. We follow the prompt engineering process described in the framework to design, test, and fine-tune the optimal prompt.

Instruction learning requires the prompt to be precise and instructional to the language model. Therefore, we initiate the prompt by setting the “Role and Context” for the model. As the
task is to extract the key financial numbers from the statement, we state the “Role and Context” as follows:

[Role and Context]: “You are an assistant who is good at extracting financial information from unstructured textual data.”

The “Role and Context” limits the model to only consider the financial information from the given information, thus restraining it from generating replies that are irrelevant to the given text. Next, to provide the general extraction requirement and set the output format, we define a set of rules for GPT-4. To illustrate, one of the optimal “Rule” sets for the Statement of Net Position is presented here:

[Rule]: “Strictly obey the following rules when extracting:
Rule 1. Find each value by recognizing the relevant row and column names.
Rule 2. If certain row or column names cannot be matched exactly, find the most likely match using fuzzy matching and surrounding information.
Rule 3. If a certain value can not be found, return '' for that value.
Rule 4. Only output the JSON format data, with key names as Liabilities, NPL, OPEB, TNP, Thousand, and Million. Do not output your analytical procedures and explanations.”

The “Rule” starts by highlighting that all the rules described below should be followed when performing each of the extractions. As the financial data are embedded in the statement, Rule 1 sets the logic for locating them by referring to the row and column names. In case any exceptions happen, such as expressive variations or missing values, Rules 2 and 3 provide relevant coping strategies. Finally, Rule 4 defines the output format of the extraction and further suppresses GPT-4 from outputting unnecessary analytical details.
Both the “Role and Context” and “Rule” prompts are to provide general guidelines on the extraction tasks. In this step, we will generate the “Task” prompt that delineates specific financial data to be extracted. This separation between different types of instructions also aligns with the logic of instruction learning, which advocates for constructing the prompts in a more clear and actionable format. Similarly, to provide one illustrative example, we provide the optimal prompt tuned for extracting the Statement of Activities:

[Task]: “The page content is a financial statement. Extract the following values from the statement:
1. Expenses for total primary government.
2. Charges for services for total primary government.
3. Operating grants and contributions for total primary government.
4. Capital grants and contributions for total primary government.
5. Total general revenues and transfers for total activities.
6. Change in net position for total activities.
7. Grants and contributions not restricted to specific programs for total activities.
8. If the values in the table are expressed in thousand, 1000. Otherwise, output ‘’.
9. If the values in the table are expressed in million, output 1000000. Otherwise, output ‘’.”

This example “Task” prompt outlines the financial data points. The prompt starts by connecting the statement (page content) with its corresponding data point description. Since each “Task” prompt is optimized for a specific statement, it is usually placed immediately after the page content to avoid misunderstanding. Specifically, items 1-7 define the data points to be extracted. The key is to describe each item as detailed as possible to eliminate multi-matching. As the values in the statement are usually expressed in thousands or millions, especially when the entity is large,
the last two items in the prompt set are to match such transformations and provide indicators for subsequent post-processing.

Prompts following the instruction learning illustrated above are all instances that ZSL can cope with. However, situations still exist where pure instructions cannot guarantee accurate extraction. Therefore we also apply FSL in some data points when the extraction needs examples to learn from. To give an illustration, we introduce the application of the FSL method on extracting “Total Long-Term Liabilities” from the Statement of Net Position. As “Total Long-Term Liabilities” usually involve identifying a list of “Long-Term Liabilities” that are all contributors to the total value, we supply several examples in the “Task” prompt. The prompt for this item is engineered as follows:

[Task]: “1. Long-Term Liabilities for total activities:
   a. Usually, they are all the line items between ‘Liability due in more than one year’ and ‘total liabilities’/‘total noncurrent liability’/’total liabilities due in more than one year’.
   b. Only extract the value for the total activities and do not miss anyone.
   c. Some example names for the line items: ‘Lease liability’, ‘Compensated absences payable’, ‘Post-closure care costs’.”

In this “Task” prompt, instead of merely providing the instructions on the tasks to be extracted, we also supply several examples of the starting and ending account names, as well as the line items to be identified. In cases where the tasks are complex or the probability of oversight is high, FSL can significantly enhance the performance model.

Although instruction learning and Zero/Few-shot Learning can guide most of the prompt engineering, we still encounter one situation during the ACFR extraction where CoT prompting
can significantly increase the accuracy. As we talked about in the TOC understanding process, GPT-4 is leveraged for identifying the corresponding pages for each target statement/section. Even though this task is not complicated, the large size of page numbers as long as their corresponding relationships will cause oversights or confusion in the model. To enhance this, we implement CoT when designing the prompt. Below is one example “Task” prompt on the TOC understanding:

[Task]: The page content is a table of contents. Identify the page numbers for each of the following statements/sections.

1. What is the first page containing the Statement of Net Position? Assign it to A.
   What is the page number/range of the immediate next statement/item following A? Assign it to B. (If the next item's page number is a range, such as "22 - 23", only keep the first number, which is 22 in this example)
   Form list_1 with A and B in [A, B] list format.

2. What is the first page containing the Statement of Activities? Assign it to C.
   What is the page number/range of the immediate next statement/item following C? Assign it to D. (If the next item's page number is a range, such as "22 - 23", only keep the first number, which is 22 in this example)
   Form list_2 with C and D in [C, D] list format.

In this prompt, we try to avoid stating all the requirements in one instruction which usually causes misunderstanding. Instead, we assign each sub-instruction in a separate expression and assign a variable to temporally save its output. With the preceding instruction being implemented, we further take the output and use it as the reference in the subsequent instruction. When all the sub-instructions are satisfied, we can easily refer to all the temperate variables and output them with a clear mapping relationship. This particular example shows how prompts can be engineered
under the guidance of CoT prompting. The FSL is also integrated into the design process to facilitate better performance.

**Batch Querying with GPT-4**

Extracting data from ACFRs typically involves handling a large number of data points and files, given that each local government issues its own annual report. To streamline this process, we integrate optimal prompt sets, specifically "Role and Context," "Rule," and "Task," into a function that can be initiated using the GPT-4 API. This function is designed to loop through all the target pages that require processing. We outline the function's logic based on the following pseudocode:  

Pseudocode for acfr_extraction function:
Function acfr_extraction takes in page_dictionary, target_page_number, model="gpt-4", api_key:
1. Retrieve the content for the given target_page_number from page_dictionary
   - page_content = content associated with key target_page_number in page_dictionary
2. Use OpenAI API to get completion
   - Set the model to "model" parameter
   - Set the API key to "api_key" parameter
   - Initialize messages with:
     - a system message that includes Role and Context, and Rule
     - a user message that includes page_content and the Task to perform
3. Send these parameters to the OpenAI.ChatCompletion.create function

---

7 Pseudocode is a simplified, language-agnostic representation of an algorithm used to outline a program's logic without the complexity of actual code.
4. Retrieve the first choice's message content from the completion response
5. Return the content of the first choice

End Function

When the function receives parameters, it executes a complete set of prompts on the page_content input to perform the extraction task. Since all ACFRs have already been converted into "page dictionaries" during the data preparation phase, and the page numbers containing the statements of interest are identified through TOC (Table of Contents) analysis, it's straightforward to map these two sets of information to acquire the content of the target pages. Finally, the function loops through each page_content to execute the prompts.

Database Construction

After extracting data using GPT-4, we temporarily store it in a Comma-Separated Values (CSV) format. While CSV is convenient for data retrieval, it is not ideal for long-term storage and querying. Since one of our objectives in extracting data from ACFRs is to build a standalone database, a standardized database format would be more suitable. With this goal in mind, we begin by preprocessing the CSV files for database construction.

The output from GPT-4 is typically plain text, even when the content appears to be numerical. Therefore, our first step is to convert all string-formatted numbers into actual numerical values that can be read and manipulated. Second, extracted financial data often contain indicators for thousands or millions, specifying whether a particular value is expressed in thousands or millions in the original statement. We transform these values back to their original units and remove the indicators. Finally, as some governments fail to disclose certain values in the ACFRs, we replace any empty fields with zeros.

Electronic copy available at: https://ssrn.com/abstract=4567607
Once the original numerical values have been accurately preprocessed in the CSV files, we proceed to transform these files into a relational database schema. To accomplish this, we employ Pentaho Data Integration, a specialized analytics and business intelligence (ABI) software. The field names within the database are aligned with the 19 corresponding field names delineated in the original CSV files. Furthermore, attention must be given to the inclusion of a unique identifier for effective data retrieval. In this context, we opt for a composite key consisting of "County Name" and "Year" to enable the selection of individual rows.

VII. EXPERT EVALUATION

It is crucial to conduct thorough and frequent evaluations during the framework design process. The evaluation involves two key steps. First, we assess the effectiveness (accuracy) and efficiency of the data extracted using our framework. In the initial test of our design, we achieved an accuracy rate of 96.1%. Subsequent updates to the prompts led to a remarkable improvement, reaching 100% accuracy in the second test. To validate these results, we compared them with data extracted manually by human experts. The results are presented in Table 1 and demonstrate that the proposed framework tremendously improves data extraction efficiency. The substantial improvement in accuracy can be attributed to our efficient use of the GPT model API and prompt engineering.

Second, during the framework’s design phase, we maintain regular communication with the GFOA to ensure that any modifications to our framework remain feasible. Also, we hope to keep professional boards like GFOA updated, thus facilitating their ability to guide data preparers toward more efficient LLM data extraction processes.
VIII. DISCUSSION

We explore the concept of extracting information from unstructured data, specifically focusing on financial reports presented in PDF format. The central objective of this study aligns closely with that of XBRL tags. XBRL’s taxonomy facilitates the standardization of financial statement structures, but it necessitates the categorization of numerical data in a predefined manner. XBRL later introduced the concept of "extension taxonomy," granting firms the ability to define tags that align better with their unique business models. While this flexibility enhances adaptability, it also introduces the potential for misuse of this mechanism. In contrast, our LLM-based method offers a solution by transferring the responsibility of data labeling (pre-processing) to data post-processing. We posit that this framework holds promise for achieving more accurate and efficient information extraction from lengthy PDF-format documents across diverse contexts.

Recognizing the existence of plugins and commercially available PDF chatbots, some of which may possess the potential to achieve similar objectives, we contend that these tools often lack the requisite capability to accurately extract information from tables. We hypothesize that even when such tools are augmented with LLMs, their proficiency may not be sufficient to achieve a high level of accuracy, primarily attributable to the broadened scope of search across all pages within the PDF document. To empirically assess this hypothesis, we conduct an additional test utilizing ChatPDF\(^8\), an industry-leading PDF chatbot equipped with LLM capabilities. This test involved employing the same set of prompts, which are applied to two ACFRs. The outcomes of our investigation indeed corroborate our initial prediction, with ChatPDF failing all four data

---

\(^8\) See [https://www.chatpdf.com/](https://www.chatpdf.com/) for more information. The website did not explicitly tell which LLM it uses to process the uploaded PDFs.
extraction tasks, as demonstrated in Appendix 2. Further research is needed in this area to enhance the effectiveness of data extraction directly from the entire document.

Our framework is not exempt from certain limitations. In comparison to the XBRL tagging method, our approach lacks the provision of a direct one-to-one mapping between extracted data and a clearly defined taxonomy. Consequently, users may encounter challenges in interpreting the results of data extraction, which may appear less straightforward in nature. Regulators may also raise objections and display a predisposition towards "algorithm aversion" concerning more intricate and "black box"-like extraction processes that demand specialized expertise and deeper analytical reasoning (Commerford et al. 2022). Nevertheless, we hold the expectation that our framework serves as a foundational platform upon which diverse pathways can be constructed, and with the accumulation of additional empirical evidence, the concern can be mitigated.

IX. CONCLUSION

We introduce a comprehensive framework for the extraction of information from unstructured textual documents in PDF format, employing state-of-the-art LLM technology. LLMs are harnessed to address the challenge of extracting and analyzing data from documents lacking predefined labels or standardized descriptions. Our study showcases the application of this framework in the context of extracting financial data from ACFRs of local governments, ultimately achieving a remarkable accuracy rate of 100% through the refinement of prompts for the GPT model, all accomplished within a notably short timeframe.

This paper demonstrates the feasibility of adopting LLMs for the extraction of accounting data, offering several noteworthy contributions to the existing literature. Firstly, we present a systematic workflow using LLMs for the efficient and effective extraction of data, characterized by high generalizability and minimal technical hurdles. Secondly, we highlight the potential of
LLMs as an alternative to data standardization, diverging from the prevalent approach of implementing a centralized and detailed taxonomy, such as the ongoing efforts to introduce XBRL in the domain of governmental accounting in the United States. Thirdly, we provide a sample illustrating the design of prompts to maximize the capabilities of LLMs in achieving accurate data extraction. This proposed framework possesses the versatility to be applied in various contexts where numerical information is presented within PDF documents. We encourage future researchers to explore the applicability of our framework in diverse domains and settings, beyond the scope of our current study.
REFERENCES


Electronic copy available at: https://ssrn.com/abstract=4567607


FIGURE 1 Flowchart of the Framework

Note: This figure presents the flowchart of the framework. The steps are detailed in the arrows, and the inputs and outputs for each step are outlined. Steps that utilize LLM are shaded in gray.
FIGURE 2 Illustration of TOC understanding

<table>
<thead>
<tr>
<th>TABLE OF CONTENTS</th>
<th>INTRODUCTORY SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Letter of Transmittal</td>
<td>1</td>
</tr>
<tr>
<td>GFOA Certificate of Achievement for Excellence in Financial Reporting</td>
<td>15</td>
</tr>
<tr>
<td>Organizational Chart</td>
<td>16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FINANCIAL SECTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>Independent Auditor’s Report</td>
</tr>
<tr>
<td>Management’s Discussion and Analysis (Unaudited Required Supplementary Information)</td>
</tr>
<tr>
<td>Basic Financial Statements:</td>
</tr>
<tr>
<td>Government-wide Financial Statements:</td>
</tr>
<tr>
<td>Statement of Net Position</td>
</tr>
<tr>
<td>Statement of Activities</td>
</tr>
</tbody>
</table>

Note: This figure shows an illustration of TOC understanding.
| Summary of Comparison of GPT-4 and Human Experts |
|-----------------------------|---------------------|---------------------|

<table>
<thead>
<tr>
<th></th>
<th>GPT-4 - initial test</th>
<th>GPT-4 - refined prompts</th>
<th>Experts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total Count of Data Points</td>
<td>152&lt;sup&gt;a&lt;/sup&gt;</td>
<td>152</td>
<td>152</td>
</tr>
<tr>
<td>Actual Count of Correct Data Points</td>
<td>146</td>
<td>152</td>
<td>150</td>
</tr>
<tr>
<td>% Correct Data Extraction</td>
<td>96.1%&lt;sup&gt;b&lt;/sup&gt;</td>
<td>100%</td>
<td>98.7%</td>
</tr>
<tr>
<td>% Absolute Variance (on Average)</td>
<td>0.03%</td>
<td>0%</td>
<td>5.2%</td>
</tr>
<tr>
<td>Total Time to Extract Data (in minutes)</td>
<td>8</td>
<td>8</td>
<td>200</td>
</tr>
<tr>
<td>PDF Conversion Time</td>
<td>4</td>
<td>4</td>
<td>NA</td>
</tr>
<tr>
<td>Code Running Time</td>
<td>4</td>
<td>4</td>
<td>NA</td>
</tr>
</tbody>
</table>

Notes: This table shows the performance evaluation result of our framework for financial data extraction tasks.

<sup>a</sup>In total, 152 data points were collected from eight ACFRs for the year 2022, each from a different county.

<sup>b</sup>The accuracy rate reached 96.1% with the initial prompt design and later improved to 100% after refining the prompts. However, we anticipate that the accuracy rate may decrease when the framework is tested on new ACFRs or under different settings.

Electronic copy available at: https://ssrn.com/abstract=4567607
APPENDIX 1 Variable List for the Illustration on the ACFR

<table>
<thead>
<tr>
<th>Variables</th>
<th>Statement</th>
<th>Account Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Assessed Value</td>
<td>Assessed Value of Taxable Property</td>
<td>Fiscal Year Ended June 30 2022, Assessed Value Total</td>
</tr>
<tr>
<td>2 General Fund Unassigned</td>
<td>Balance Sheet Governmental Funds</td>
<td>Fund Balances Unassigned, General</td>
</tr>
<tr>
<td>3 General Fund Assigned</td>
<td>Balance Sheet Governmental Funds</td>
<td>Fund Balances Assigned, General</td>
</tr>
<tr>
<td>4 Total Assets</td>
<td>Balance Sheet Governmental Funds</td>
<td>Assets - Total Assets, Total</td>
</tr>
<tr>
<td>5 Cash and Investments</td>
<td>Balance Sheet Governmental Funds</td>
<td>Assets: (Cash and Investments; Restricted Cash and Investments), General</td>
</tr>
<tr>
<td>6 Long-Term Liabilities for Governmental Activities</td>
<td>Note 10. Long-Term Obligations</td>
<td>Governmental Activities - Total Governmental Activities, Balance June 30, 2022</td>
</tr>
<tr>
<td>7 Program Revenues</td>
<td>Statement of Activities</td>
<td>Total Primary Government, Program Revenues (Charges for Service; Operating Grants and Contributions)</td>
</tr>
<tr>
<td>8 General Revenues and Transfers</td>
<td>Statement of Activities</td>
<td>Total General Revenues and Transfers, Net (Expenses) Revenues and Changes in Net Position Total</td>
</tr>
<tr>
<td>9 Expenses</td>
<td>Statement of Activities</td>
<td>Total Primary Government, Expenses</td>
</tr>
<tr>
<td>10 Operating Grants and Contribution</td>
<td>Statement of Activities</td>
<td>Total Primary Government, Program Revenues Operating Grants and Contributions</td>
</tr>
<tr>
<td>11 Capital Grants and Contributions</td>
<td>Statement of Activities</td>
<td>Total Primary Government, Program Revenues Capital Grants and Contributions</td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Statement Type</td>
</tr>
<tr>
<td>---</td>
<td>------------------------------------------------------------------------------</td>
<td>----------------------------------------</td>
</tr>
<tr>
<td>12</td>
<td>Unrestricted Aid Reported with General Revenue</td>
<td>Statement of Activities</td>
</tr>
<tr>
<td>13</td>
<td>Change in Net Position</td>
<td>Statement of Activities</td>
</tr>
<tr>
<td>14</td>
<td>Unrestricted Net Position</td>
<td>Statement of Net Position</td>
</tr>
<tr>
<td>15</td>
<td>Long-Term Liabilities</td>
<td>Statement of Net Position</td>
</tr>
<tr>
<td>16</td>
<td>Net Pension Liability</td>
<td>Statement of Net Position</td>
</tr>
<tr>
<td>17</td>
<td>OPEB Liabilities</td>
<td>Statement of Net Position</td>
</tr>
<tr>
<td>18</td>
<td>GF Expenditures</td>
<td>Statement of Revenues, Expenditures, and Changes in Fund Balances Governmental Funds</td>
</tr>
<tr>
<td>19</td>
<td>Operating Revenues</td>
<td>Statement of Revenues, Expenditures, and Changes in Fund Balances Governmental Funds</td>
</tr>
</tbody>
</table>
APPENDIX 2 Test of ChatPDF’s Accuracy on ACFRs Extraction

Notes: These two screenshots show the results when using ChatPDF to extract financial data from the ACFRs of the County of Los Angeles and the County of Tulare.