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

Acknowledgment: We thank Diane Janvrin (Editor), Uday Murthy (discussant), two anonymous reviewers, Michael Alles, Irfan Bora, Soohyun Cho, Shayne Kavanagh, Alexander Kogan, participants of the 58th, 59th, and 61st World Continuous Auditing & Reporting Symposiums, the 2023 Rutgers Governmental Accounting & Auditing Update Conference, the 2023 Bright Internet Global Symposium, the 2024 PCAOB Seminar, the 2024 Missouri Society of CPA Governmental Accounting Conference, the 2024 Joint Midyear Meeting of the AIS and SET Sections, and the 2024 *Journal of Information Systems* Conference. We also thank Buhe Li for research assistance. We gratefully acknowledge the support from the Government Finance Officers Association (GFOA). The opinions expressed are those of the authors and do not reflect the position of the GFOA. Any errors are solely the responsibility of the authors.

JEL Classifications: M41; O31; C81

Keywords: Data extraction; Information processing; Unstructured data; Large Language Model (LLM); ChatGPT; PDF Reports; Design Science; Accounting information systems

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 introduces a novel framework for automated financial data extraction from PDF-formatted files. Following a design science methodology, this research develops the framework through a combination of text mining and prompt engineering techniques. The framework is subsequently applied to analyze governmental annual reports and corporate ESG reports, which are presented in PDF format. Test results indicate that the framework achieves an average 99.5% accuracy rate in a notably short time span when extracting key financial indicators. A subsequent large out-of-sample test reveals an overall accuracy rate converging around 96%. This study contributes to the evolving literature on applying LLMs in accounting and offers a valuable tool for both academic and industrial applications.

I. INTRODUCTION

Accounting is a study primarily rooted in the quantitative expression of economic phenomena (Davidson 1966). The advent of machine-readable datasets, notably Compustat in 1962, revolutionized the accessibility of financial data for all stakeholders, thereby accelerating research in accounting and capital markets (Teoh 2018). However, despite the availability of existing structured datasets, vast realms of financial data remain sequestered in unstructured formats. These include but are not limited to fiscal indicators embedded within Portable Document Format (PDF)-formatted governmental Annual Comprehensive Financial Reports (ACFRs) (Li, Wei, Moffitt, and Vasarhelyi 2023), prices and quantity negotiated within contracts (Yan and Moffitt 2019), and greenhouse gas (GHG) emission figures disclosed in environmental, social, and governance (ESG) reports (Jiang, Gu, and Dai 2023). The significant amount of granular financial data embedded in such unstructured sources has impeded regulatory processes (Li et al. 2023), academic research (Warren, Moffitt, and Byrnes 2015), and investment decisions (El-Haj, Alves, Rayson, Walker, and Young 2020).

The advancements in large language models (LLMs) at the end of 2022 have ushered in a promising avenue to transcend these constraints by converting human-generated, unstructured information into machine-readable, standardized databases (Gu, Schreyer, Moffitt, and Vasarhelyi 2023; Vasarhelyi, Moffitt, Stewart, and Sunderland 2023). Unlike existing text mining techniques, such as sentiment analysis, topic modeling, or keyword-based frequency counting, which typically perform aggregative summarization of text, LLMs excel in accurately isolating and retrieving granular information for in-depth analysis. This level of precision was previously unattainable using conventional text mining techniques (Gu et al. 2023). LLMs can be likened to college students who can comprehend user questions and perform various textual tasks with significantly higher efficiency (Ng 2023). In light of this, our study embarks on pioneering an LLM-enabled framework aimed at extracting financial data from unstructured sources, offering invaluable insights for market participants, policymakers, and researchers.

Specifically, we have designed an LLM-enabled framework adept at processing PDF-formatted files, isolating and retrieving predefined financial data therein. The framework incorporates data preparation, prompt engineering, batch querying, and database construction. To validate its performance, we applied the framework to two accounting scenarios, including extracting key indicators from the local governments' ACFRs and corporations' ESG reports. We subsequently benchmarked their accuracy and efficiency against manual methods. Although only validated on two accounting scenarios, the framework is designed to be generally applicable to the majority of PDF-formatted data sources.

We follow the design science research (DSR) approach (Geerts 2011; Gregor and Hevner 2013; Hevner, March, Park, and Ram 2004; Peffers, Tuunanen, Rothenberger, and Chatterjee 2007) to develop and validate the artifact. It is developed in response to the pervasive challenge stakeholders face when obtaining financial data from unstructured sources (Issa 2018; Jiang et al.

2023; Yan and Moffitt 2019), as existing methods fall short of effectively resolving it. Specifically, we adopt the six-step DSR methodology proposed by Peffers et al. (2007). Geerts (2011) finds Peffers et al.'s (2007) framework suitable for DSR in the accounting information system domain due to its comprehensive objectives and consistency with the previous DSR. Unlike David, Dunn, McCarthy, and Poston's (1999) Research Pyramid, which focuses on generating research questions, Peffers et al.'s (2007) framework provides a comprehensive guide for the DSR process. Although Gregor and Hevner (2013) offer a similar framework, it lacks real case demonstrations using previous DSR literature, a gap that Peffers et al.'s (2007) framework addresses. More recently, Myers, Snow, Waddoups, and Wood (2024) recommend Peffers et al.'s (2007) framework as the guideline for conducting DSR in accounting, based on their review of 51 DSR papers.

Initially, we identify the prevailing challenges posed by unstructured data in conveying financial information (*Identify Problem and Motivation*). Subsequently, we define the objectives for addressing the problem, which encompass validating the effectiveness of the framework and enhancing its efficiency in data extraction (*Define Objectives of a Solution*). Next, we detail the development of the framework (*Design and Development*) and illustrate its application using two real-world data extraction tasks (*Demonstration*). To evaluate and refine the framework, we validate the extracted results with the actual values presented in the documents and solicit expert insights (*Evaluation*). Finally, we document both the framework and its applications in this manuscript (*Communication*). TABLE 1 summarizes the development of the framework.

In the first demonstration, we applied the framework to extract financial indicators from U.S. local governments' ACFRs. The initial test results revealed an accuracy rate of 96.1% when matched against the actual values from the reports. Upon closer inspection, we observed that the inaccuracies were caused by reasons such as omitted data, misjudgment of units, and misidentification of rows or columns. Consequently, we refined the corresponding prompts following the prompt engineering techniques introduced in the framework. The second test achieved a 100% accuracy rate. In terms of efficiency, both tests conducted using the framework required less than 4% of the time that was originally needed with the manual method.

In the second application, we tested the framework's ability to extract key performance indicators from the ESG reports of companies listed on the Hong Kong Stock Exchange (HKEX). The initial test resulted in a 93.3% accuracy rate with the primary source of error being the misjudgment units. Upon further improvement to the prompts, the second test generated an accuracy rate of 98.9%. Similarly, the framework was 9-18 times faster than the human manual extraction process.

One concern regarding the evaluation process is that the framework has only been applied to ACFRs and ESG reports with small in-sample extractions, which may not reflect its performance on a large-scale out-of-sample group. To address this and ensure that the framework's performance is not biased due to overfitting, we conducted tests on more than 4,000 county-year ACFRs spanning from 2013 to 2021, resulting in over 80,000 data extractions, and achieved an average accuracy rate of 96%.

This research makes several contributions to the literature. First, we devise and validate a framework that leverages LLMs to extract financial data from unstructured sources. Researchers across diverse disciplines can utilize our framework to access novel data, allowing them to either expand their existing research pipelines or venture into unexplored research domains. For instance, the demonstration of our framework provides a solution to the data availability limitation in governmental accounting by extracting key financial indicators from ACFRs in PDF format, facilitating research in various public domains (W. Kim, Plumlee, and Stubben 2022). Second, the proposed framework contributes to the burgeoning body of research examining 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). Our findings shed light on the potential of LLMs as an alternative approach to traditional costly data standardization methods. Such an approach is notably different from the widely adopted method of employing a centralized and comprehensive taxonomy, as exemplified by the intense efforts to incorporate eXtensible Business Reporting Language (XBRL) into governmental accounting within the U.S. Instead of creating a predefined taxonomy, the new LLM-enabled framework allows data preparers to continue formulating reports in a traditional way, while simultaneously allowing the users to access and analyze data in a modern, automated manner. Third, the proposed framework echoes the call for accounting research that should be relevant to practice. The accounting literature has long been criticized for its perceived detachment from real-world concerns (Burton, Summers, Wilks, and Wood 2022; McCarthy 2012; Rajgopal 2021; Waymire 2012). Our framework serves as a solution to address one of the practical problems faced by accounting stakeholders in utilizing unstructured data (Balderas 2021; CPA Canada and AICPA 2020; GFOA 2023a, 2023b), thereby amplifying its relevance to practice.

The remainder of this paper is structured as follows: Section II introduces the research background and reviews relevant literature. Section III outlines the motivations for developing the artifact, while Section IV further sets the objectives of the proposed framework. The framework is developed in Section V and illustrated in Section VI. Section VII presents the evaluation results, and Section VIII shows additional tests. Finally, Section IX discusses and concludes the paper.

II. BACKGROUND AND LITERATURE REVIEW

Development of LLM

The field of artificial intelligence (AI) has long sought to develop machines capable of reasoning and thought. Various attempts have been made to understand human natural language, with LLMs emerging as one of the most recent and remarkable developments. The inaugural LLM, known as the “Transformer” model, was introduced by 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-enabled AI tools to complement or augment human workers in various domains. It is worth noting that LLMs are not limited to question-answering, exam-taking, or creative storytelling. For instance, OpenAI’s most recent foundation model can handle diverse tasks, including the analysis of visual inputs (Liu, Li, Wu, and Lee 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. During this phase, the model is typically trained on a massive dataset to learn the underlying structures and nuances of the language. Once a model is pre-trained, it can be used off the shelf, much like typing a question (a “prompt”) to the ChatGPT dialog interface.

LLM Applications in Accounting

LLMs have been applied to understand financial information from textual disclosure data. Huang, Wang, and Yang (2023) leveraged techniques from the Bidirectional Encoder Representations from Transformers (BERT) model, one of the precedents of GPT, to develop a novel LLM known as FinBERT. This model can identify sentiments and topics in financial texts more effectively than traditional, dictionary-based tools. Advanced models like GPT-3 or GPT-4 offer even further improvements in sentiment analysis capabilities and can be used to generate stock return predictions based on news headlines (Lopez-Lira and Tang 2023).

The generative capability of LLM also offers unique advantages. For instance, A. Kim, Muhn, and Nikolaev (2023) created a new measure, "*Bloat,*" calculated as the length of a summary provided by ChatGPT-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 in accounting, using the latest GPT-4 model for demonstration.

Despite these pioneering applications of LLMs in the accounting literature, information extraction using LLMs remains underexplored. A concurrent paper by Gu et al. (2023) represents one of the early studies proposing the application of LLMs in auditing. While they suggest that qualitative information, such as pension plan details from annual reports, can be extracted using LLMs from an auditing perspective, they do not provide a detailed framework specifically designed for extracting information from unstructured sources. Additionally, their study does not discuss how to extract quantitative information, such as figures in financial statements. Quantitative information extraction is different from qualitative information extraction. While both tasks require contextual understanding, accuracy is paramount when dealing with numbers. 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). Further research is necessary to bridge this gap.

III. PROBLEM IDENTIFICATION AND MOTIVATION

The prevalence of unstructured data in the accounting domain is on the rise due to the dynamic nature of the business environment (Alaamer, Jumaa, Alqashar, and Wadi 2023; Balderas 2021). A predominant format of this data is PDF documents, such as ESG reports, financial reports of U.S. local governments, non-public company financial disclosures, analyst reports, or Uniform Bank Performance Reports. A significant challenge identified in both accounting literature and practice is the extraction of financial information from these unstructured PDF documents (CPA Canada and AICPA 2020; Bernard and Rao 2021; Issa 2018; Jiang et al. 2023; Li et al. 2023; Yan and Moffitt 2019). Existing text mining methods are limited to aggregative summarization of text, thus falling short in effectively obtaining granular data (Senave, Jans, and Srivastava 2023; Bochkay, Brown, Leone, and Tucker 2023). Reviews of the literature indicate that manual data collection remains the primary method for retrieving financial data from unstructured sources (Altamuro, Gray, and Zhang 2022; Beardsley, Mayberry, and McGuire 2021; Beck 2018). This challenge becomes particularly acute in the face of documents characterized by inconsistent formatting and item descriptions (El-Haj et al. 2020; Li et al. 2023), thereby hindering regulatory processes, research development, and investment decisions.

Facing these increasing challenges, this study aims to develop an LLM-enabled framework for extracting financial data from PDF documents, particularly in cases where machine-readable formats or well-developed taxonomies are absent. We believe that recent advancements in LLM technology hold the potential to markedly improve the comprehension and retrieval of financial data. Our goal is to bridge the existing gaps in data accessibility across various domains by implementing this framework, thereby facilitating better access to crucial financial information.

IV. OBJECTIVES OF THE FRAMEWORK

Given that current data collection methods from unstructured sources are predominantly manual and labor-intensive (Altamuro et al. 2022; Beardsley et al. 2021; Beck 2018), we opt to leverage the LLM-enabled framework to improve both the effectiveness and efficiency of data extraction. To address the challenge of inconsistent financial terminology, which complicates accurate information identification, our aim is to ensure that the extracted data accurately reflects the values presented in the source documents. Furthermore, we seek to enhance the efficiency of the data extraction process by implementing batch processing and developing a function for streamlined operation.

V. FRAMEWORK DESIGN

In this section, we describe the methodologies and processes that form the foundation of our framework. As illustrated in FIGURE 1, the framework involves the preparation of 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 these processes. 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 with subsequent data preprocessing and batch querying. To achieve this format transformation, one can leverage software, Python packages, or online web-based tools, depending on the source documents and specific tasks. The choice of method should be justified through a comparative assessment of multiple solutions.[1]

[1] Software options include Adobe Acrobat, PDFElements, Able2Extract Pro, among others. For Python programming, packages such as PyPDF2, Tika, and PDFminer are available. Additionally, web-based tools like CloudConvert, Smallpdf, and pdf2go offer convenient online solutions. Based on the authors' comparison of the methods mentioned

Ideally, the selected method should generate consistent conversion performance.

After conversion, we outline three steps to break down files into granular chunks that are most relevant to the target data, thereby enhancing the accuracy and consistency of subsequent LLM processing. These steps include Table of Contents (TOC) Understanding, Page Range Refinement, and Page Dictionary Establishment.

For the lengthy documents with a TOC, we first feed them into LLM for TOC Understanding. This step leverages the LLM's natural language understanding capabilities, along with specifically designed prompts, to accurately interpret the TOC.[2] The goal is to segment the documents into distinct content sections, organizing them by page numbers. As the TOC is prepared by human beings, it offers an efficient and accurate way to identify target pages from the entire file. However, the page ranges parsed from the TOC may still contain an extensive number of pages because the TOC does not always provide detailed segmentations, such as in the case of the Notes section of an annual financial report, which often spans more than 50 pages. Therefore, we refine those page ranges to a more granular level, focusing on the pages most relevant to the target data points (Page Range Refinement). This refinement serves to further minimize the search range, resulting in more accurate and consistent outputs. To achieve this, we first observe the target data points and compile a list of keywords for each item. Next, we establish item-specific rules, such as the minimum number of digits required on a page, or keywords that should and should not appear on the target page (e.g., the keyword "MD&A" typically does not appear in the balance

above, Able2Extract Pro demonstrates the highest overall accuracy in converting PDF files containing embedded tables and scanned figures.

[2] Since many PDF files lack a readily recognizable TOC using programming language, we have developed a more consistent method to let LLM recognize the TOC. The design of the prompt is introduced in the prompt engineering step.

sheet table, whereas "balance" does). Finally, we employ Python regular expressions (regex)[3] to represent these keywords and digits and develop Python programs to filter out irrelevant pages. After the above two steps, we subsequently transform the entire converted plain text file into a Python dictionary, where the page number serves as the "Key" and its relevant page content serves as the "Value."[4] This step establishes a well-defined mapping between page numbers and the corresponding page content, allowing for easy retrieval of page content based on the page number.

After completing these three steps, it is straightforward to obtain target page content based on the combination of Refined Target Page Number/Range and Page-Dictionary, as shown in FIGURE 1. For instance, a target page identified with the page number *N* from the TOC Understanding and Page Range Refinement can be directly retrieved from the Page Dictionary by searching for the "Key" of *N*; the corresponding "Value" of *N* will be the target page content.[5]

For shorter files or files without a TOC, we typically skip the TOC Understanding and Page Dictionary Establishment, focusing only on Page Range Refinement. In these instances, the entire file can be considered as a page range for the application of this method, as illustrated in the second case presented in Section VI. The content of the refined pages/ranges will be directly retained after eliminating irrelevant pages.

Prompt Engineering

With the source data prepared, the subsequent step involves designing the prompts to be input into the LLM. Prompts are natural language instructions or queries that humans use to

[3] Python regular expressions are a set of characters and symbols used to search, manipulate, and match patterns in strings. These are used here to match the rules (keywords) surrounding the target data points.

[4] A Python dictionary is a mutable, unordered collection of key-value pairs, where each unique key is associated with a value, allowing for efficient retrieval and updating of data based on keys. We use Python regex to identify the page numbers. See examples in Section VI.

[5] In dictionary format, the "Key" is the identifier of the data, and the "Value" is the data itself. Under the context of this method, the "Key" is the page number and the "Value" is the page content.

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 et al. 2023; Min et al. 2022). Within our framework, we employ *instruction learning*, *zero/few-shot learning*, and *Chain-of-Thought prompting* to perform prompt engineering.

Instruction Learning

Extracting data from unstructured sources requires clear and consistent instructions to prevent errors or omissions (Li et al. 2023). To achieve this, we first apply *instruction learning* to design the prompt. *Instruction learning* is a method that is designed to enhance the performance and generalization capabilities of LLMs by utilizing 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.[6] 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 task and describes the context. By contextualizing the LLM, it can narrow its focus to a specific

[6] Given that the foundation model is pre-trained to understand human languages, minor adjustments in the phrasing of these instructions are permissible, provided that the intended meaning remains consistent. For instance, users can say either "Identify and list all the dates and event names mentioned in the text" or "Extract and enumerate the dates and names of events found in the text." However, the actual utility of the instructions depends on the trial and testing by users.

knowledge domain and reasoning logic, thus enhancing response accuracy and consistency.

Rule: This set of prompts should outline the requirements that the LLM must adhere to throughout the entire extraction process. These requirements include, but are not limited to, the data searching logic (e.g., matching row and column names in tables or searching context in paragraphs), alternative strategies for handling exceptions if the target information is not present (e.g., return "NaN"), and the desired output format of the extracted data (e.g., JSON format). These prompts should serve as the general principles guiding the extraction process in the desired direction. They should clearly articulate a fixed set of operations for extracting these items (akin to how a human being would perform to obtain these items), thereby avoiding errors such as hallucinations and inconsistent results in two identical attempts.[7] The specific rules outlined here should be user-defined, tailored to the items being extracted.

Task: The third set of prompts contains detailed descriptions of each extraction item. It is essential to specify each item to be extracted from the unstructured source. However, achieving the optimal description necessitates manual iteration and evaluation. Generally, maintaining a high level of detail is essential to ensure accurate information extraction. A rule of thumb is that the prompt should be developed in such a manner that it conveys the task unambiguously and is sufficiently clear to be understood by a fresh college graduate (Ng 2023).

Zero/Few-shot Learning

Zero-shot learning highlights the foundation model's ability to address tasks not encountered during its training phase. This method enables LLMs to process new and unseen data by leveraging their extensive pre-existing knowledge base, thus eliminating the need for explicit training on specific tasks (examples) (Kojima, Gu, Reid, Matsuo, and Iwasawa 2022). In terms of

[7] Hallucination in LLMs refers to the generation of information that lacks grounding in their training data or the input they received, often described as "making things up."

data extraction, LLMs can address some tasks using contextual cues without the need for examples, making *zero-shot learning* particularly well-suited for straightforward tasks that do not require complex analytical processing, such as identifying a product's price within a contract. Employing this method, we design prompts for some simple tasks by clearly outlining the requirements for each one in the "Task" prompts, without providing examples.

However, some extraction tasks are not straightforward, such as those requiring the identification of multiple sub-items to derive a total value. This is where *few-shot learning* steps in. A modern development in deep learning and natural language processing, *few-shot learning* involves instructing LLMs using a minimal number of examples (Brown et al. 2020; Zhao, Wallace, Feng, Klein, and Singh 2021).[8] This approach aids the model in understanding and inferring new instances based on examples. *Few-shot learning* is particularly effective in unique extraction scenarios where training data is sparse. When implementing this approach, rather than merely including descriptions in "Task," one should integrate several examples along with the descriptions to facilitate accurate extraction.

Chain-of-Thought Prompting

Another engineering technique suggested in this framework is *Chain-of-Thought prompting* (Wei et al. 2022), a recent advancement in prompt engineering that is specifically designed to enhance the reasoning capabilities of LLMs. Specifically, *Chain-of-Thought 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). *Chain-of-Thought prompting* has consistently demonstrated an improved model performance, especially in tasks demanding detailed reasoning.

[8] The exact number of examples may vary depending on trial and testing. Typically, providing two to five examples that represent the major variations is sufficient to create an effective prompt.

To deploy *Chain-of-Thought prompting* effectively across various accounting domains, it is crucial to emulate human problem-solving methods by breaking complex tasks into smaller, sequential steps. This modular approach ensures that each step is straightforward for LLM to perform, and the output from one step can seamlessly serve as the input for the next. Domain-specific knowledge, terminologies, and best practices can be supplied in certain steps to enhance the performance.[9]

It is also worth noting that various prompt engineering techniques can be combined to elicit better results. For instance, *Chain-of-Thought prompting* can be effectively integrated with *few-shot learning* during the prompt engineering process. One can start prompting by including a few well-chosen examples, followed by a Chain-of-Thought-style breakdown that demonstrates the reasoning process used to obtain that data point. The underlying logic is that *few-shot learning* essentially primes the model with a basic understanding of the extraction task through examples, while *Chain-of-Thought prompting* enhances its performance through a step-by-step reasoning process. Such a dual approach can enrich an LLM's ability to handle complex accounting tasks with enhanced precision and contextual relevance.

Batch Querying with LLM

Typical methods in the accounting and auditing literature that apply LLMs rely on the user interface for interaction (Emett, Eulerich, Lipinski, Prien, and Wood 2023; Eulerich and Wood

[9] To illustrate, suppose the task at hand is to conduct a financial risk analysis for a company considering a major investment. Instead of directly assessing whether the investment is risky, we simplify the complex task by breaking it down into manageable, sequential steps. Initially, we have the LLM identify the investment and its objectives to set the foundation for the analysis. Following this, we prompt the LLM to assess the company's financial health using its key financial indicators, evaluate the current market and economic conditions, and analyze the potential returns and risks of the investment using financial models. This step might involve domain-specific instructions teaching LLM how to conduct the evaluation based on the user's experience. If the task is outside of the user's domain, a general rule is to instruct the LLM by conveying the step unambiguously and making it sufficiently clear to be understood and performed by a fresh college graduate, as documented in the *instruction learning* section. Finally, we ask the LLM to synthesize the findings from each step to compile a comprehensive analysis and make a reasoned recommendation on the investment.

2023; Föhr, Schreyer, Juppe, and Marten 2023; Gu et al. 2023). The benefits of this approach are evident, such as enhanced interpretability, straightforward demonstration, and a user-friendly, no-code environment (Li and Vasarhelyi 2024). However, data extraction tasks frequently require sequential analysis of numerous target data points within a file. Thus, manually inputting each prompt through a user interface is not efficient, especially when dealing with massive documents. To streamline this process, we integrate the prompts into a function that can be readily applied using LLMs' Application Programming Interface (API).

Specifically, we formalize the designed three sets of prompts, namely "Role and context," "Rule," and "Task," into a Python function with the parameters including API Key, model name, data source, and output location. After initiating the process, the function automatically iterates through each file in the data source and executes the defined data extraction tasks in the prompts.

This approach offers several benefits. First, it facilitates ease of deployment. Users, irrespective of their programming expertise, can easily engage with it by supplying the API key, specifying data source, and starting the process. Second, our approach simplifies maintenance and migration. In case of errors or updates, the Python function can be easily traced, enabling quick debugging and enhancement. Furthermore, this functional formatting facilitates the seamless application to other extraction scenarios and LLMs. Instead of a complete process redesign, the framework can be deployed to other data sources featuring distinct data points through reassignment of the data source and adjustment of the data preparation and prompt engineering.[10]

Database Construction

Upon completing the batch querying, the extracted data will be consolidated within a

[10] Even though some adjustments are required for the framework to be applied to other scenarios, the data preparation and prompt engineering approaches introduced in Section V can be used to guide those adjustments (see the second application in Section VI).

Database Management System to enable long-term storage and effortless querying.[11] The choice of a Database Management System 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 illustrative Database Management System. In line with the criteria for selecting a Database Management System, the method for transferring contents from a CSV file (output format of LLM) into the database should also be chosen.[12]

VI. ILLUSTRATIONS OF THE FRAMEWORK

This section demonstrates two applications of the framework. Following the procedures detailed in Section V, we applied the framework to extract financial data from the ACFRs of U.S. local governments and ESG reports of HKEX-listed companies. TABLE 2 Panel A summarizes the structural differences between ACFRs and ESG reports, and Panel B compares the artifact steps between them. Selecting two application scenarios from different sources and capital markets underscores the versatility and general applicability of the framework.

ACFR of the U.S. Local Governments

The PDF-formatted ACFRs prepared by individual governments are the primary source of a comprehensive set of financial information for U.S. local governments. Since 1970, ACFRs have become the national paradigm for local governmental reporting for over half a century (HandWiki 2022). Other than ACFRs, there is no centralized, electronic, and publicly accessible database of

[11] Prior to constructing the database, it is necessary to preprocess the data extracted by 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.

[12] One might utilize Python or opt for established software for the transformation process. Python packages include psycopg2, SQLAlchemy, and pandas with SQLAlchemy, among others. Software options encompass Pentaho, pgAdmin, Navicat, and more.

governmental financial data (W. Kim et al. 2022). In addition to the non-machine-readable barriers introduced by the PDF format, the description for the same item can vary substantially among different entities. This lack of readily available financial data for local governments significantly limits the research in governmental accounting (Issa 2018). Consequently, extracting financial data from the ACFRs to create a structured database not only presents an ideal opportunity to demonstrate our framework but also makes significant contributions to both researchers and practitioners (GFOA 2023a, 2023b; Granof 2020).

We began the demonstration by consulting the experts from the Government Finance Officers Association (GFOA) to compile a list of key financial data to be extracted. As a professional association of national-wide local government financial officers, the GFOA acknowledges the limitations of accessing large-scale accounting data from ACFRs. As a result, it has advocated applying technology to enhance the extraction of financial data to support decision-making (GFOA 2023a, 2023b). The list of 19 financial variables to be extracted is provided in Appendix 1. All of them were evaluated and confirmed by the accountants and experts from GFOA. Following the GFOA's recommendation, we randomly selected eight local counties from California to implement the framework. This choice was based on California's significant representation among local governments in the U.S. and its consistently high reporting quality.

Data Preparation

The first step involves data conversion, as all the ACFRs of U.S. local governments are presented in PDF format. We used the software "Able2Extract Pro" to batch convert all the PDF files into plain text format. Given that the software integrates Optical Character Recognition (OCR) and other basic page manipulation capabilities, it could automatically recognize information from the scanned pages and restore the pages with a 90-degree rotated layout.

As the ACFRs are usually lengthy with TOC, 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 applied one of the industry-leading LLMs, namely GPT-4, to read and understand the TOC.[13] To read the page range of the target statement, as illustrated in FIGURE 2, we leveraged *few-shot learning* and first prompted GPT-4 to identify the starting page of the "Statement of Net Position" for the County of Orange, which is Page 41. Next, we prompted it to identify the beginning of the immediate next distinct statement, which starts on Page 43. Then we can logically deduce that the "Statement of Net Position" encompasses both Pages 41 and 42.[14]

Next, we performed Page Range Refinement on sections that occupy an extended range of pages, including the Notes to Basic Financial Statement (Notes). For instance, "Long-Term Obligations for Governmental Activities" is typically located within the Notes section. Direct input of the entire Notes section into GPT-4 leads to significant temporal inefficiencies and can also affect the precision of data retrieval.[15] Specifically, we found the target page in the Notes section usually satisfies the following three criteria: (1) The page 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.[16] Utilizing Python regex, we filtered each page in the Notes section based on these criteria, effectively narrowing down the

[13] Our manual observations indicated that the TOC section within ACFRs typically spans no more than the initial 165 lines of the converted document. Therefore, the program will input the first 200 lines of each ACFR into GPT-4 for TOC Understanding.

[14] The actual prompt used here in the TOC Understanding is illustrated in the Prompt Engineering section.

[15] Excessive input into the LLM could introduce noise, thus reducing the extraction accuracy. See more discussion in Section VII by taking the whole PDF file as input. Furthermore, longer contexts will consume more input tokens, potentially leading to higher costs in LLM service fees if it is token-based.

[16] The threshold of "30" digits was determined through a manual examination of five documents, where it was observed that pages discussing Long-Term Liabilities typically contained more than 30 digits. The validity of this threshold was further confirmed by its high accuracy rate in subsequent large-scale sample tests.

identification of "Long-Term Obligations for Governmental Activities" to merely one or two pages within the Notes section.

Finally, we followed the framework to transform the converted text file into a Page Dictionary using a set of Python codes (mainly regex). Our initial step involved dividing the document into chunks based on page breaks, identified by double empty lines as separators.[17] Recognizing that page numbers are typically located at the top or bottom of each page, we examined the first and last lines of each chunk for these numbers. In instances of formatting issues that obscured page number identification—such as encountering years or financial figures from tables mistaken for page numbers—we automatically increased the page number for the affected chunk by one, relative to the preceding chunk. In cases where no numbers could be discerned, we maintained the page number from the previous chunk.

Prompt Engineering

The combination of the refined page number/range and the Page Dictionary allowed us to retrieve the target page content. We further utilized GPT-4 to extract the 19 variables from the target page content (See Appendix 1). We followed the prompt engineering techniques described in the framework to design, test, and refine the prompt.

Instruction learning requires the prompt to be precise and instructional to the LLM. Therefore, we initiated the prompt by setting up the "Role and Context," "Rule," and "Task." Appendix 2 Panel A outlines some example prompts we used. Specifically, the "Role and Context" limits GPT-4 to only consider the financial data from the provided information, thereby preventing it from generating responses based on its own knowledge.

[17] We utilize regex to identify the page numbers within each page and isolate individual pages. Subsequently, we convert the file into a Python dictionary format, facilitating the mapping between page numbers and page content. A double empty line is the default page separator for text converted from Able2Extract Pro. Other tools may generate different separator patterns.

Next, to provide the general extraction requirement and set the output format, we defined a set of rules for GPT-4. Rule 1, given that financial data are embedded in the statement, sets the logic for locating them by referring to the row and column names. In case any exception occurs, 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 provide general guidelines on the extraction tasks. In the "Task" prompt, we delineated specific data points to be extracted following the *zero-shot learning* (prompt without examples). This separation between different types of instructions also aligns with the logic of *instruction learning*, which advocates constructing the prompts in a clearer and more actionable format. Similarly, as an illustrative example, we provide part of the optimal prompt tuned for extracting the Statement of Activities in Appendix 2 Panel A [Task].[18] The prompt starts by aligning the page content (statement) with the corresponding data points to be extracted to avoid confusion.[19] Specifically, tasks 1-3 define the data points to be extracted. 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 "Task" are to identify such transformations and provide indicators for subsequent preprocessing in the database construction stage.

In addition to *instruction learning* and *zero-shot learning*, we also applied *few-shot learning* in some data points when the model needed examples to learn from. As illustrated in

[18] The optimal prompts are determined based on the accuracy rate generated during the evaluation. Usually, the prompts are considered optimal when the accuracy rate converges without any further improvement. In this demonstration, prompts were considered optimal when they achieved a 100% accuracy rate, which took three iterations to converge. The number of iterations required may vary across different tasks. The refinements involved in our demonstration primarily focus on detailing task descriptions and the inclusion of "thousands" and "millions" as additional output fields.

[19] The "confusion" refers to the potential lack of clarity about which data points correspond to which parts of the financial statement.

Appendix 2 Panel B, we applied *few-shot learning* on extracting “Total Long-Term Liabilities” from the Statement of Net Position. As “Total Long-Term Liabilities” involves identifying a list of line items contributing to the total amount, instead of merely providing the searching logic in bullet points a and b, we also supplied several examples in bullet point c.[20] As the LLM might not be familiar with the terminologies of certain items, *few-shot learning* can enhance the overall accuracy of these extractions.

Although *instruction learning* and *zero/few-shot learning* can guide most of the prompt engineering for this extraction task, we still encountered one situation where *Chain-of-Thought prompting* significantly enhanced accuracy. When performing the TOC Understanding, GPT-4 is prompted to identify the corresponding pages for each target statement/section. However, due to the diverse formatting of TOC, the LLM sometimes failed to accurately capture the page range for each statement if no specific guidance is provided.[21] To enhance this, we implemented *Chain-of-Thought prompting* when designing the TOC Understanding prompt. As illustrated in Appendix 2 Panel C, we tried to avoid stating all the requirements in one instruction. Instead, we organized each sub-instruction in a separate expression and assigned a variable (temporary variable) to temporarily save its output. In this example, we first asked GPT-4 to find the first page containing the Statement of Net Position and assign it to a temporary variable *A*. Following the first instruction, we further took the output (*A*) and used it as the reference in the subsequent instruction (What is the page number/range of the immediate next statement/item following *A*? Assign it to *B*). When all the subsequent instructions have been satisfied, we can easily refer to all the temporary variables

[20] Through an untabulated random sampling test (10 samples), we found that the extractions were not sensitive to the examples used in the prompt, provided that the prompts clearly indicated that these were merely examples of the target data points.

[21] For example, in the TOC, one way of describing the page numbers for Statement A is “12-14,” while Statement B may be indicated as “15-16.” An alternative approach is to specify the page number of Statement A as “12” and Statement B as “15.” Without clear instructions to teaching the LLM how to identify page numbers, these variations could potentially lead to errors.

(e.g., *A* and *B* in this example) and output them in the desired format. During this process, we also provided GPT-4 with an example (enclosed in parentheses) on how to recognize page numbers expressed in a range format in the TOC, leveraging the concept of *few-shot learning*.

Batch Querying with GPT-4

We integrated the engineered prompt sets, specifically "Role and Context," "Rule," and "Task," into a function that can be initiated using the GPT-4 API. The pseudocode in Appendix 3 outlines the logic of this function.[22] When the function receives parameters (page_dictionary, target_page_number, model="gpt-4", and api_key), it executes the set of defined prompts to perform the extraction task. Since the page numbers containing the target data points are identified through TOC Understanding, and all ACFRs have been converted into Page Dictionaries during the data preparation step, the function will simply take these two sets of information to acquire the page_content. Finally, the function loops through each page_content to execute the prompts.

Database Construction

To prepare the data extracted from LLM for database construction, our initial step involves transforming numbers originally in text format from GPT-4's output into actual numerical values. This includes recalibrating financial figures to reflect their true scale by removing indicators that denote values in thousands or millions. In cases where values are missing in the ACFRs, a default value of "NaN" is assigned to these empty fields. Following this preprocessing, we proceeded to develop a relational database utilizing Pentaho Data Integration software. We implemented a composite key combining "County Name" and "Year" to uniquely identify each record.

ESG Reports from HKEX Companies

The second application involves using the framework to extract GHG emissions and

[22] Pseudocode is a simplified, language-agnostic representation of an algorithm used to outline a program's logic without the complexity of actual code.

hazardous waste data disclosed in the ESG report (See Appendix 4). Given the global emphasis on climate disclosures, the accurate capture of these key performance indicators will serve as the foundation for stakeholders to perform large-scale analysis and comparison (Krasodomska and Zarzycka 2020). For the sake of simplicity, in this application, we only document the changes from the first application and provide general guidelines on the adjustments.

Compared to ACFRs, ESG reports are much shorter and often lack a detailed TOC. Therefore, after PDF conversion, we skipped the TOC Understanding and Page Dictionary Establishment and treated the entire report as a whole for Page Range Refinement. This requires a redesign of the Python regex based on the new data points. After manual observation, we compiled a list of keywords, including "NOx," "SOx," "greenhouse gas," "GHG," "scope 1," "scope 2," and "hazardous," to construct the regex. We then used this regex to filter each page of the converted file, eliminating irrelevant pages and retaining page content that matched the regex.

Next, we followed the "Role and Context," "Rule," and "Task" structure to adjust the prompt for the new data source. For instance, in the "Role and Context," we informed LLM to search for climate-related information within the ESG report. In the "Rule," as the emission data might not be disclosed in the table, we adjusted the search method to match both the row and column names and the paragraph context. In addition, we rewrote each individual description of the task.[23] At the end of the prompt, using *few-shot learning*, we furnished two example extractions by presenting the original emission description from the ESG report and the desired output. Finally, we used the same function with GPT-4 to perform the batch query as in the first demonstration and further relied on the company's stock number and year as the primary key to form the database.

In general, as shown above, transforming the framework from one application to another

[23] One example of the task is "Find the amount of Air emission NOx for the current year."

data source only requires some adjustments to the data preparation and prompt engineering processes. As the approaches introduced in the framework cater to broader PDF data sources, one can skip certain steps that are not necessary for their applications. The modification of the prompts can also be guided by *instruction learning*, *zero/few-shot learning*, and *Chain-of-Thought prompting*.

VII. EVALUATION

ACFR of the U.S. Local Governments

The evaluation of the ACFR application encompasses two aspects. Qualitatively, we consulted one senior research manager and one analyst from the GFOA to gather expert assessments on the framework. Given that these GFOA members were not involved in the design of the framework, their assessments are both professional and objective. The experts unanimously concurred that the framework is effective and efficient in extracting financial data from unstructured sources, offering a significant improvement over their previous methodology, which primarily depends on manual effort. To further refine the framework, the experts suggested providing additional examples in the prompts for complex items to enhance the learning process. Additionally, they encouraged us to explore advanced tools to enhance the OCR processing of scanned PDFs, as older files are typically presented in scanned format. Following that, we updated some of the prompts with more examples and upgraded the software to improve the OCR process.[24]

Quantitatively, we evaluated the framework by calculating the accuracy and efficiency of the extraction. Accuracy is gauged by determining the percentage of data points aligned with the

[24] For instance, to improve the extraction of Long-Term Liabilities, which involves identifying sub-items, we supplied several examples of Long-Term Liabilities to enhance the accuracy. The latest version of Able2Extract Pro (version 19) has enhanced OCR performance for scanned PDFs compared to previous versions.

actual value presented in the data source. The results presented in TABLE 3 show that the framework generates high extraction accuracy. In the initial test, we achieved an accuracy rate of 96.1%. Upon further examination, we identified several issues related to the errors, including omissions when the LLM was instructed to extract a list of line items, misjudgment of units (such as thousands or millions), and incorrect identification of rows and columns.

To resolve this, we began revising the prompts and prepared a second test. This step is particularly important in the DSR as it facilitates the development of the artifact by iterating back to the design phase to refine the process (Peffers et al. 2007). For instance, to address the omissions during extraction, we introduced a more deterministic approach in prompt engineering by specifying reference keywords in the data source that indicate which data points should be included. Additionally, to resolve the misjudgment of units, we redesigned the prompt to include the unit as an additional field to be returned in the output.[25] Finally, we provided more detailed descriptions of row and column names to prevent misidentification. The updates to the prompts led to a remarkable improvement, reaching 100% accuracy rate in the second test. This improvement can be attributed to our efficient prompt modification under the guidance of the framework.

It is also noteworthy that our comparison revealed inaccuracies within the expert-derived results. Specifically, a prevalent error identified among human analysts' results was the direct replication of numerical values from the statement through a "copy and paste" method, without adequate conversion of these figures from their thousand or million denominations back to their fundamental values.

[25] For instance, consider a scenario where the data to be extracted is the total expenditure, shown as $500 on the balance sheet. In our evaluation, we found that if the total expenditure is expressed in thousands on the balance sheet, the LLM output could be either $500 or $500,000. To enhance accuracy, we can prompt the LLM to always extract the value presented in the table (which, in this case, is $500), and return an additional field indicating whether the values in this statement are expressed in thousands or not.

In terms of efficiency, the framework demonstrated a substantial enhancement over manual processes. On average, the framework reduced the duration of the process from 200 minutes to merely 8 minutes before updating the prompts, representing a 25-fold increase in efficiency compared to experts' manual extraction.

In general, currently there is no one-size-fits-all prompt that can fit all extraction tasks, and it is also hard to obtain the optimal prompt on the initial attempt. The best practice is to start with the simple prompt following the prompt engineering techniques and evaluate the initial results. Based on errors, one should adjust the prompts by either adding more examples, deleting irrelevant descriptions, or rewriting in a different way. After several rounds of modifications and evaluation, the prompt that yields the highest accuracy rate and most consistent results should be the best fit for the task.

ESG Reports from HKEX Companies

To evaluate the application on ESG reports, two of the authors manually performed the same extraction tasks and then averaged the results for comparison with the framework (refer to results in TABLE 4). In the initial test, the framework achieved an accuracy rate of 93.3%, with most errors related to misjudging units between grams, kilograms, and tonnes. Following the same modification in the first demonstration, we introduced additional fields to the data points and had LLM extract the units as separate output fields. The second test resulted in an accuracy rate of 98.9%, with one instance encountering a computational error during the summation of sub-items.[26] Comparatively, humans achieved 100% accuracy rate in the extraction task, given that ESG reports are relatively shorter and easier for manual searching than ACFRs.

In terms of efficiency, the framework required approximately 2.5 to 5 minutes to complete

[26] We expect that this type of computational (mathematical) error will be reduced with future evolvements of LLM.

the extraction tasks, depending on whether the PDF conversion is performed or not. This is a nine-fold to 18-fold increase in efficiency compared to the manual extraction process, which took about 45 minutes to finish. Overall, the application on ESG reports further validates the framework's generalizability to various extraction scenarios with different variables.

VIII. ADDITIONAL TESTS

In this section, we performed several additional tests to further enhance the relevance of the framework and demonstrate its performance over a large sample and with different LLMs. First, the framework is proposed for extracting financial information from unstructured sources, with a specific focus on PDF-formatted reports. While acknowledging the existence of plugins and commercially available PDF chatbots that may aim to achieve similar objectives, we argue that these tools frequently lack the capability to accurately extract information from tables in lengthy PDF reports. We hypothesize that such tools fall short of achieving high accuracy due to the challenges posed by the comprehensive search required across all pages of the PDF document. To empirically test this hypothesis, we conducted an additional test using the whole PDF question and answer (Q&A) functionalities of GPT-4 and Claude 2. This test involved applying the same set of prompts used by the framework directly to the entire ACFRs. The results confirmed our initial prediction. Tests with GPT-4's full PDF Q&A function consistently produced errors after several rounds conducted at different times. Conversely, Claude 2 was unable to process the request, hindered by the extensive length of the PDF file. Further research is needed in this area to improve the effectiveness of data extraction directly from entire documents.

Second, the framework has demonstrated high accuracy in extracting financial data across two scenarios. However, there is a potential concern regarding its performance on large-scale extraction tasks, as developing and testing the prompts on the same set of documents might lead to overfitting. To address this issue, we undertook an untabulated test, applying the prompts initially developed from 8 ACFRs to a more extensive dataset of over 4,000 county-year ACFRs. This extensive application resulted in the extraction of more than 80,000 data points. An examination through random sampling revealed that the framework maintained an average accuracy rate of 96%, further underscoring its effectiveness.[27]

Finally, instead of exclusively using GPT-4 as the illustrated LLM, we also compared the performance of GPT-4, Claude 2, and Gemini on a random sample of files. In the untabulated test, GPT-4 achieved an average accuracy rate of 96.8%, and Claude 2 achieved 93.7%. Gemini had the lowest accuracy rate at 69%.[28] However, we expect that future versions of LLMs will demonstrate improved performance as they undergo continuous refinement.

IX. DISCUSSION AND CONCLUSION

This study demonstrates the feasibility of using LLMs for extracting accounting data, making significant methodological contributions to both academia and practice (CPA Canada and AICPA 2020; Bernard and Rao 2021; Issa 2018; Li et al. 2023). In addition, we illuminate the potential of LLMs as an alternative to traditional data standardization methods. For instance, the first application of our framework closely aligns with the principles outlined in the Financial Data Transparency Act (FDTA), which aims to enhance the accessibility and understanding of financial data reported by governmental agencies. While XBRL has been widely regarded as a top choice for FDTA compliance (Bauguess 2018), it also has certain limitations, such as misuse of custom extensions. In contrast, our LLM-based framework offers a solution by transferring the

[27] The random sampling evaluation involved randomly selecting 85 ACFRs, corresponding to 1,615 data points, for manual evaluation of accuracy. Based on the results of this large-scale out-of-sample test, we expect the accuracy of the extraction to remain consistent in the long run, assuming no major changes in report formats. Given the random errors observed during the large sample test, we also expect the accuracy to be enhanced if the performance of LLM is further improved.

[28] The tests on the GPT-4 and Claude 2 were performed in January of 2024 and Gemini was in April of 2024.

responsibility of data pre-processing (labeling) to data post-processing (LLM extraction). This holds promise for achieving more accurate and efficient information extraction from lengthy PDF-format documents across diverse contexts, in line with the FDTA's objectives.

However, the framework is not exempt from certain limitations. In comparison to the XBRL, 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 extracted data as it can be less intuitive. Regulators may also raise objections and exhibit a predisposition towards "algorithm aversion" in relation to more intricate and "black box"-like extraction processes (O'Leary 2023), which require specialized expertise and deeper analytical reasoning (Commerford, Dennis, Joe, and Ulla 2022).[29] Future research should explore ways to implement traceability within the framework, enabling researchers or practitioners to understand how data points are identified, supported by intrinsic evidence from the context. This enhancement could also improve the credibility and verifiability of the extracted results. Another limitation is data privacy and security when dealing with financial information using the LLM (O'Leary 2023). It is crucial to ensure that sensitive information is not leaked while being processed by LLMs. A forward-looking solution to this potential risk involves the analysis of sensitive data on an offline LLM platform. A last limitation of this framework is the financial cost associated with using LLMs like GPT-4. Within our framework, the batch query requires the use of an API that incurs a token-based expense. To mitigate this cost constraint, one could consider alternatives like GPT-3.5 or Facebook's LLaMA, which are typically more affordable or even free of charge.[30] We also anticipate that the costs associated with using LLM will decrease over time due to the maturation

[29] However, we have not observed any objections from the GFOA regarding the use of LLMs in data extraction.

[30] By December 2023, the GPT-4 API price was $0.03 per 1k tokens for input (prompt) and $0.06 per 1k tokens for output (completion). The GPT-3.5-turbo API price was $0.0015 per 1k tokens for input (prompt) and $0.002 per 1k tokens for output (completion). LLaMA is open-sourced and free.

of the technology and advancements in computing power.

Overall, we anticipate that this framework will serve as a foundational platform, enabling the development of diverse applications. With the accumulation of further empirical evidence and technological advancements, it is expected that concerns surrounding this approach can be effectively mitigated. This proposed framework possesses the versatility to be applied in various contexts, such as the Official Statement of municipal bond issuance, which is only available in PDF format. It is also applicable in corporate settings, including legal contracts or documents from legacy systems that are only available in PDF format, where a local LLM or an encrypted version (e.g., GPT-4 Enterprise) can be employed to securely process these internal documents. We encourage future research to explore the applicability of this framework in diverse domains and settings beyond the scope of the current study.

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work the authors used ChatGPT in order to improve language and readability. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

REFERENCES

Alaamer, A., M. Jumaa, K. Alqashar, and R. A. Wadi. 2023. Management accounting in the era of big data. In *Emerging Trends and Innovation in Business and Finance*, edited by R. El Khoury and N. Nasrallah, 869–879. Singapore: Springer Nature.

Altamuro, J. L. M., J. V. Gray, and H. Zhang. 2022. Corporate integrity culture and compliance: A study of the pharmaceutical industry. *Contemporary Accounting Research* 39 (1): 428–458. https://doi.org/10.1111/1911-3846.12727

Balderas, D. V. 2021. The Rise of Unstructured Data. *Cloudera Blog*. https://blog.cloudera.com/the-rise-of-unstructured-data/

Bauguess, S. W. 2018. The Role of Machine Readability in an AI World. SEC Speeches and Statements. Available at: https://www.sec.gov/news/speech/speech-bauguess-050318

Beardsley, E. L., M. A. Mayberry, and S. T. McGuire. 2021. Street versus GAAP: Which effective tax rate is more informative? *Contemporary Accounting Research* 38 (2): 1310–1340. https://doi.org/10.1111/1911-3846.12651

Beck, A. W. 2018. Opportunistic financial reporting around municipal bond issues. *Review of Accounting Studies* 23 (3): 785–826. https://doi.org/10.1007/s11142-018-9454-2

Bernard, R. N., and A. Rao. 2021. Automating Data Extraction with AI. *PwC*. Available at: https://www.pwc.com/gx/en/issues/data-and-analytics/artificial-intelligence/publications/ai-automation-data-extraction.html

Bochkay, K., S. V. Brown, A. J. Leone, and J. W. Tucker. 2023. Textual analysis in accounting: What's next? *Contemporary Accounting Research* 40 (2): 765–805. https://doi.org/10.1111/1911-3846.12825

Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, et al. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems* 33: 1877–1901. https://doi.org/10.48550/arXiv.2005.14165

Burton, F. G., S. L. Summers, T. J. Wilks, and D. A. Wood. 2022. Relevance of accounting research (ROAR) scores: Ratings of titles and abstracts by accounting professionals. *Accounting Horizons* 36 (2): 7–18. https://doi.org/10.2308/HORIZONS-2020-147

Chung, H. W., L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, et al. 2022. Scaling instruction-finetuned language models. (Working paper). https://doi.org/10.48550/arXiv.2210.11416

Commerford, B. P., S. A. Dennis, J. R. Joe, and J. W. Ulla. 2022. Man versus machine: Complex estimates and auditor reliance on artificial intelligence. *Journal of Accounting Research* 60 (1): 171–201. https://doi.org/10.1111/1475-679X.12407

Chartered Professional Accountants (CPA) Canada, and American Institute of Certified Public Accountants (AICPA). 2020. The Data-Driven Audit: How Automation and AI are Changing the Audit and the Role of the Auditor. https://us.aicpa.org/content/dam/aicpa/interestareas/frc/assuranceadvisoryservices/downloadabledocuments/the-data-driven-audit.pdf

David, J. S., C. L. Dunn, W. E. McCarthy, and R. S. Poston. 1999. The research pyramid: A framework for accounting information systems research. *Journal of Information Systems* 13 (1): 7–30. https://doi.org/10.2308/jis.1999.13.1.7

Davidson, S. 1966. Editor's preface. *Journal of Accounting Research* 4: iii–iii. https://www.jstor.org/stable/2490161

de Kok, T. 2023. Generative LLMs and textual analysis in accounting: (Chat)GPT as research assistant? (Working paper). https://doi.org/10.2139/ssrn.4429658

Dowling, C., and S. A. Leech. 2014. A Big 4 firm's use of information technology to control the audit process: How an audit support system is changing auditor behavior. *Contemporary Accounting Research* 31 (1): 230–52. https://doi.org/10.1111/1911-3846.12010

El-Haj, M., P. Alves, P. Rayson, M. Walker, and S. Young. 2020. Retrieving, classifying and analysing narrative commentary in unstructured (glossy) annual reports published as PDF files. *Accounting and Business Research* 50 (1): 6–34. https://doi.org/10.1080/00014788.2019.1609346

Emett, S. A., M. Eulerich, E. Lipinski, N. Prien, and D. A. Wood. 2023. Leveraging ChatGPT for enhancing the internal audit process – A real-world example from a large multinational company. (Working paper). https://doi.org/10.2139/ssrn.4514238

Eulerich, M., and D. A. Wood. 2023. A demonstration of how ChatGPT can be used in the internal auditing process. (Working paper). https://doi.org/10.2139/ssrn.4519583

Föhr, T. L., M. Schreyer, T. A. Juppe, and K.-U. Marten. 2023. Assuring sustainable futures: Auditing sustainability reports using AI foundation models. (Working paper). https://doi.org/10.2139/ssrn.4502549

Gao, L., A. Madaan, S. Zhou, U. Alon, P. Liu, Y. Yang, J. Callan, and G. Neubig. 2023. PAL: Program-aided language models. *Proceedings of the 40th International Conference on Machine Learning*, 10764–10799. Available at: https://proceedings.mlr.press/v202/gao23f.html

Geerts, G. L. 2011. A design science research methodology and its application to accounting information systems research. *International Journal of Accounting Information Systems* 12 (2): 142–151. https://doi.org/10.1016/j.accinf.2011.02.004

Government Finance Officers Association (GFOA). 2023a. Financial Intelligence through Artificial Intelligence. Available at: https://www.gfoa.org/materials/gfr1223-finance-ai

Government Finance Officers Association (GFOA). 2023b. Navigating the Next Frontier: AI's Role in Reshaping Local Governance. Available at: https://www.gfoa.org/events/int101223

Granof, M. H. 2020. Envisioning the future of government reporting: Looking back to move forward. *The CPA Journal* 90 (10/11): 38–42.

Gregor, S., and A. R. Hevner. 2013. Positioning and presenting design science research for maximum impact. *MIS Quarterly* 37 (2): 337–355. https://www.jstor.org/stable/43825912

Gu, H., M. Schreyer, K. Moffitt, and M. A. Vasarhelyi. 2023. Artificial intelligence co-piloted auditing. (Working paper) https://doi.org/10.2139/ssrn.4444763

HandWiki. 2022. Social: Comprehensive annual financial report. Available at: https://handwiki.org/wiki/Social:Comprehensive_annual_financial_report

Hevner, A. R., S. T. March, J. Park, and S. Ram. 2004. Design science in information systems research. *MIS Quarterly* 28 (1): 75–105. https://doi.org/10.2307/25148625

Hodge, F. D., K. I. Mendoza, and R. K. Sinha. 2021. The effect of humanizing robo-advisors on investor judgments. *Contemporary Accounting Research* 38 (1): 770–792. https://doi.org/10.1111/1911-3846.12641

Huang, A. H., H. Wang, and Y. Yang. 2023. FinBERT: A large language model for extracting information from financial text. *Contemporary Accounting Research* 40 (2): 806–841. https://doi.org/10.1111/1911-3846.12832

Issa, H. 2018. AIS research and government accounting research compared: Special section of

JETA on the use of AIS technology in government reporting. *Journal of Emerging Technologies in Accounting* 15 (1): 103–106. https://doi.org/10.2308/jeta-10590

Jiang, L., Y. Gu, and J. Dai. 2023. Environmental, social, and governance taxonomy simplification: A hybrid text mining approach. *Journal of Emerging Technologies in Accounting* 20 (1): 305–325. https://doi.org/10.2308/JETA-2022-041

Kim, A. G., M. Muhn, and V. V. Nikolaev. 2023. Bloated disclosures: Can ChatGPT help investors process information? (Working paper). https://doi.org/10.2139/ssrn.4425527

Kim, A. G., and V. V. Nikolaev. 2023. Context-based interpretation of financial information. (Working paper). https://doi.org/10.2139/ssrn.4317208

Kim, W. J., M. A. Plumlee, and S. R. Stubben. 2022. Overview of U.S. state and local government financial reporting: A reference for academic research. *Accounting Horizons* 36 (3): 127–148. https://doi.org/10.2308/HORIZONS-18-158

Kojima, T., S. Gu, M. Reid, Y. Matsuo, and Y. Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems* 35 (December): 22199–22213.

Krasodomska, J., and E. Zarzycka. 2020. Key performance indicators disclosure in the context of the EU directive: When does stakeholder pressure matter? *Meditari Accountancy Research* 29 (7): 1–30. https://doi.org/10.1108/MEDAR-05-2020-0876

Küster, S., T. Steindl, and M. Goettsche. 2023. The informational content of key audit matters: Evidence from using artificial intelligence in textual analysis. (Working Paper). https://doi.org/10.2139/ssrn.4464713

Li, H., and M. A. Vasarhelyi. 2024. Applying large language models in accounting: A comparative analysis of different methodologies and off-the-shelf examples. (Working paper). https://ssrn.com/abstract=4650476

Li, H., D. Wei, K. Moffitt, and M. A. Vasarhelyi. 2023. Addressing the "last mile problem" in open government data: A framework for data extraction from PDF-type governmental reports. (Working paper). https://doi.org/10.2139/ssrn.4385883

Liu, H., C. Li, Q. Wu, and Y. J. Lee. 2023. Visual instruction tuning. *Advances in Neural Information Processing Systems* 36. https://doi.org/10.48550/arXiv.2304.08485

Lopez-Lira, A., and Y. Tang. 2023. Can ChatGPT forecast stock price movements? Return predictability and large language models. (Working paper). https://doi.org/10.2139/ssrn.4412788

McCarthy, W. E. 2012. Accounting craftspeople versus accounting seers: Exploring the relevance and innovation gaps in academic accounting research. *Accounting Horizons* 26 (4): 833–843. https://doi.org/10.2308/acch-10313

Min, S., X. Lyu, A. Holtzman, M. Artetxe, M. Lewis, H. Hajishirzi, and L. Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? (Working paper). https://doi.org/10.48550/arXiv.2202.12837

Myers, N., M. Snow, N. Waddoups, D. A. Wood. 2024. Suggestions for producing and reviewing design science research in accounting. (Working paper).

Ng, A. 2023. What LLMs can and cannot do. Coursera. Available at: https://www.coursera.org/learn/generative-ai-for-everyone/lecture/VYXx5/what-llms-can-and-cannot-do

O'Leary, D. E. 2023. Enterprise large language models: Knowledge characteristics, risks, and organizational activities. *Intelligent Systems in Accounting, Finance and Management* 30 (3): 113–119. https://doi.org/10.1002/isaf.1541

OpenAI. 2022. Introducing ChatGPT. Available at: https://openai.com/blog/chatgpt

Peffers, K., T. Tuunanen, M. A. Rothenberger, and S. Chatterjee. 2007. A design science research methodology for information systems research. *Journal of Management Information Systems* 24 (3): 45–77. https://doi.org/10.2753/MIS0742-1222240302

Rajgopal, S. 2021. Integrating practice into accounting research. *Management Science* 67 (9): 5430–5454. https://doi.org/10.1287/mnsc.2020.3590

Reynolds, L., and K. McDonell. 2021. Prompt programming for large language models: Beyond the few-shot paradigm. *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*, 1–7. CHI EA '21. https://doi.org/10.1145/3411763.3451760

Senave, E., M. J. Jans, and R. P. Srivastava. 2023. The application of text mining in accounting. *International Journal of Accounting Information Systems* 50 (September): 100624. https://doi.org/10.1016/j.accinf.2023.100624

Sun, T., and M. A. Vasarhelyi. 2018. Embracing textual data analytics in auditing with deep learning. *International Journal of Digital Accounting Research* 18. https://doi.org/10.4192/1577-8517-v18_3

Teoh, S. H. 2018. The promise and challenges of new datasets for accounting research. *Accounting, Organizations and Society* 68–69 (July): 109–117. https://doi.org/10.1016/j.aos.2018.03.008

Vasarhelyi, M. A., K. C. Moffitt, T. Stewart, and D. Sunderland. 2023. Large language models: An emerging technology in accounting. *Journal of Emerging Technologies in Accounting* 20 (2): 1–10. https://doi.org/10.2308/JETA-2023-047

Vaswani, A., N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* 30. https://doi.org/10.48550/arXiv.1706.03762

Warren, J. D., Jr, K. C. Moffitt, and P. Byrnes. 2015. How big data will change accounting. *Accounting Horizons* 29 (2): 397–407. https://doi.org/10.2308/acch-51069

Waymire, G. B. 2012. Introduction for essays on the state of accounting scholarship. *Accounting Horizons* 26 (4): 817–19. https://doi.org/10.2308/acch-50236

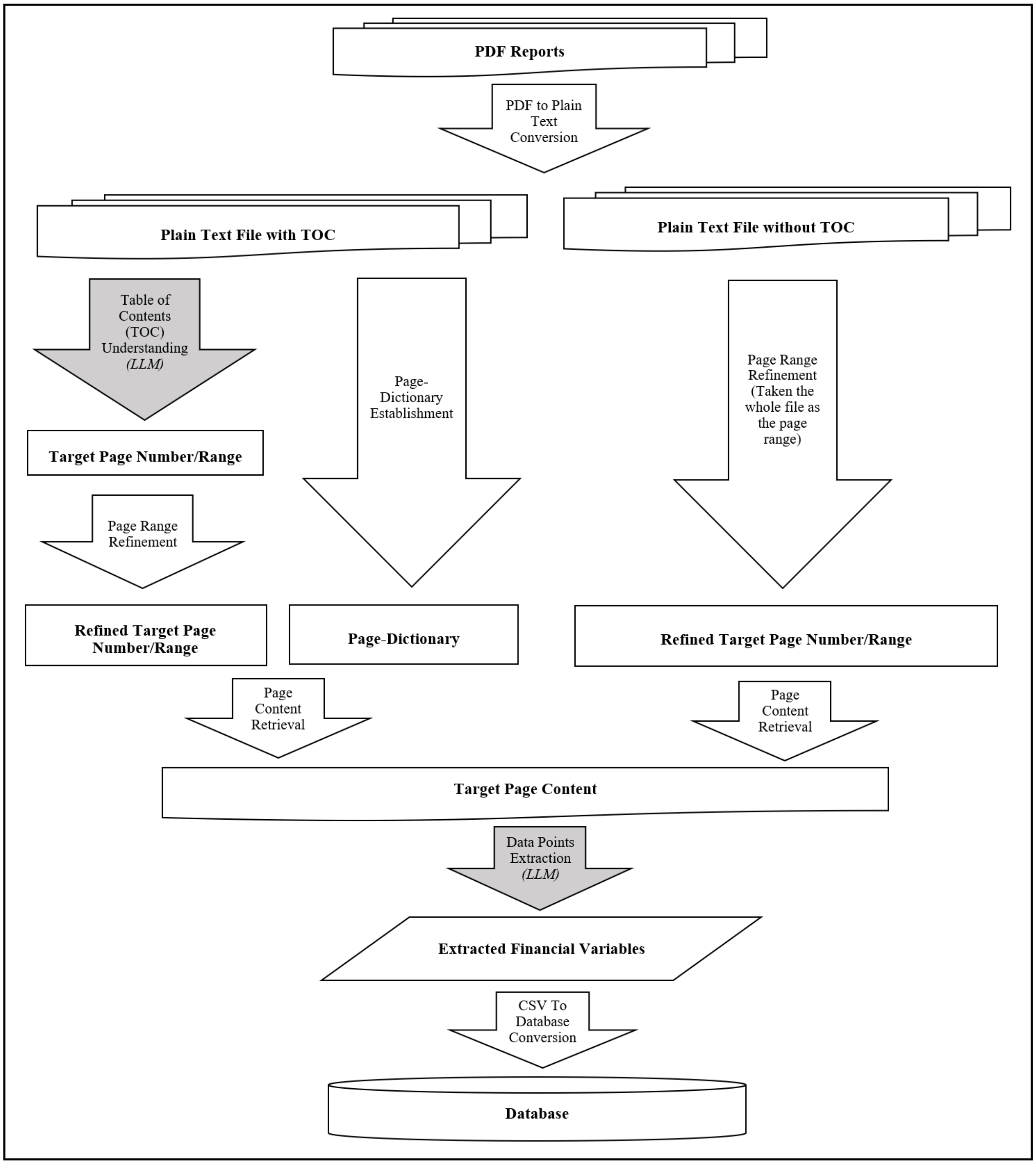
Wei, J., X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems* 35 (December): 24824–24837. https://doi.org/10.48550/arXiv.2201.11903

Wu, C., and R. B. Dull. 2021. Accessing cloud data to expand research and analytical opportunities: An example using IRS/AWS data for nonprofit organizations. *Journal of Emerging Technologies in Accounting* 18 (2): 171–183. https://doi.org/10.2308/JETA-18-12-29-28

Yan, Z., and K. C. Moffitt. 2019. Contract analytics in auditing. *Accounting Horizons* 33 (3): 111–126. https://doi.org/10.2308/acch-52457

Zhao, Z., E. Wallace, S. Feng, D. Klein, and S. Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *Proceedings of the 38th International Conference on Machine Learning*, 12697–706. PMLR. Available at: https://proceedings.mlr.press/v139/zhao21c.html

FIGURE 1 Flowchart of the Framework


PDF Reports
PDF to Plain Text Conversion
Plain Text File with TOC
Plain Text File without TOC
Table of Contents (TOC) Understanding (LLM)
Page-Dictionary Establishment
Page Range Refinement (Taken the whole file as the page range)
Target Page Number/Range
Page Range Refinement
Refined Target Page Number/Range
Page-Dictionary
Refined Target Page Number/Range
Page Content Retrieval
Page Content Retrieval
Target Page Content
Data Points Extraction (LLM)
Extracted Financial Variables
CSV To Database Conversion
Database

Note: This figure presents the flowchart of the framework. The arrows detail the steps, along with the corresponding inputs and outputs before and after the arrows. Steps that utilize LLM are shaded in gray. The PDF reports originating from various data sources are supplied by the end users.

FIGURE 2 Illustration of TOC Understanding

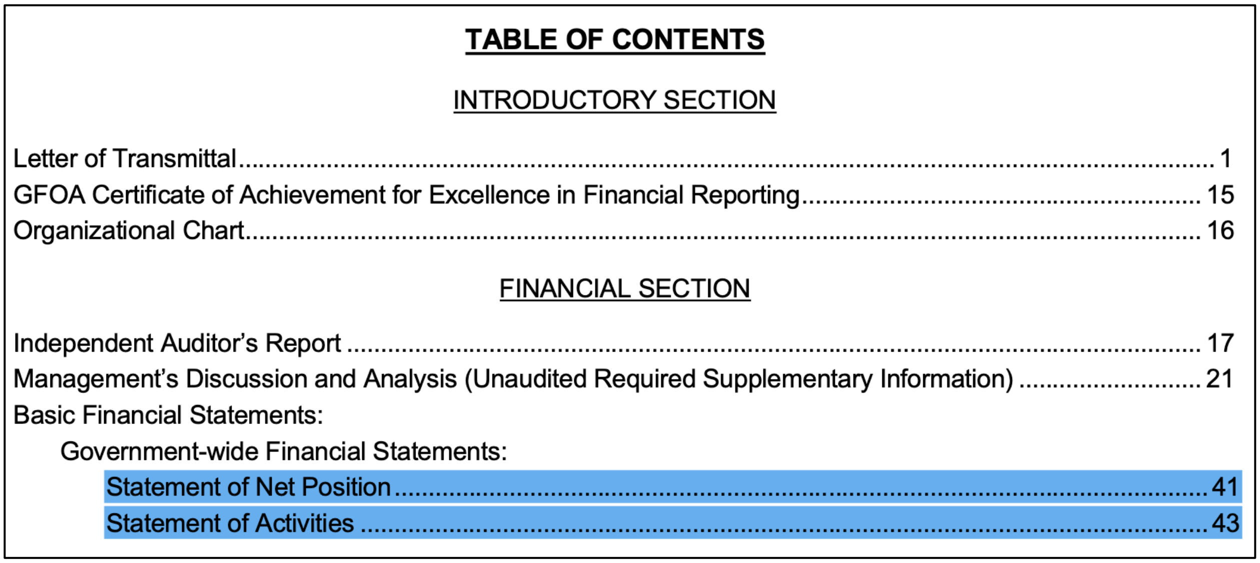


TABLE OF CONTENTS

INTRODUCTORY SECTION

Letter of Transmittal 1
GFOA Certificate of Achievement for Excellence in Financial Reporting 15
Organizational Chart 16

FINANCIAL SECTION

Independent Auditor's Report 17
Management's Discussion and Analysis (Unaudited Required Supplementary Information) 21
Basic Financial Statements:
Government-wide Financial Statements:
Statement of Net Position 41
Statement of Activities 43

Note: This figure shows an illustration of TOC Understanding. Source: County of Orange, CA, 2021 Annual Comprehensive Financial Report (ACFR).

TABLE 1 Development of the Framework Based on Peffers et al.'s (2007) Process

Steps in Peffers et al.'s (2007) DSR Process	Application of Steps in the Development of the Framework
Identify Problem and Motivation	Identify the problem of vast amounts of financial data being sequestered in unstructured formats, which impedes regulatory processes, academic research, and investment decisions.
Define the Objectives of a Solution	Improve both the effectiveness and efficiency of data extraction from unstructured sources.
Design and Development	Design an LLM-enabled framework incorporating data preparation, prompt engineering, batch querying, and database construction to extract financial data from unstructured sources.
Demonstration	Demonstrate the framework by extracting key indicators from local governments' ACFRs and corporations' ESG reports.
Evaluation	Evaluate the efficiency and accuracy of the extraction process and results from the ACFRs and ESG reports.
Communication	Document both the framework development and its demonstrations in this manuscript.

Note: This table summarizes the development of the framework based on Peffers et al.'s (2007) process.

TABLE 2 Comparing Artifact Steps Between ACFRs and ESG Reports

Panel A: Structural Difference Between ACFRs and ESG Reports

Structural Difference	**ACFRs**	**ESG Reports**	**Extraction Strategy**
TOC presentation	All ACFRs include a TOC	Only some ESG Reports include a TOC	Perform TOC Understanding for ACFRs. Skip TOC Understanding for ESG reports
Length	Range between 100-400 pages	Usually fewer than 70 pages	Perform Page Dictionary Establishment for ACFRs. Skip Page Dictionary Establishment for ESG reports and perform the Page Range Refinement directly on the entire ESG report

Panel B: Comparing Artifact Steps Between ACFRs and ESG Reports

Category	**Step in the Framework**	**ACFRs Illustration**	**ESG Reports Illustration**
Data Preparation	PDF to Plain Text Conversion	Converting ACFR to plain text	Converting ESG report to plain text
Data Preparation	TOC Understanding	Obtain page number/range based on LLM interpreting the Table of Contents	N/A due to the usual lack of a TOC and the relative brevity of the ESG reports
Data Preparation	Page Range Refinement	Refine certain page ranges to eliminate irrelevant pages using Python regex	Refine the whole ESG report to eliminate irrelevant pages using Python regex
Data Preparation	Page Dictionary Establishment	Convert the plain text into a Python Dictionary	N/A due to the usual lack of a TOC and the relative brevity of the ESG reports
Data Preparation	Page Content Retrieval	Combine the refined page number/range and Page Dictionary to obtain the page content	Obtain the page content from the refined pages
Prompt Engineering and Batch Querying with LLM	Data Points Extraction (LLM)	Design prompts using prompt engineering techniques to extract data	Design prompts using prompt engineering techniques to extract data
Database Construction	CSV to Database Conversion	Transform the extracted data into a database	Transform the extracted data into a database

Note: Panel A summarizes the key structural differences between ACFRs and ESG reports. Panel B compares the artifact steps between ACFRs and ESG reports.

TABLE 3 Illustration Results on ACFRs

	GPT-4 - initial test	**GPT-4 - refined prompts**	**Experts**
Total Count of Data Points	152	152	152
Actual Count of Correct Data Points	146	152	150
% Correct Data Extraction	96.1%	100%	98.7%
Total Time to Extract Data (in minutes)	8	4	200
PDF Conversion Time	4	NA	NA
Code Running Time	4	4	NA

Note: This table shows the performance evaluation results of the framework for financial data extraction tasks on ACFRs. A total of 152 data points were collected from 8 ACFRs for the year 2022, each from a different county. The accuracy rate reached 96.1% with the initial prompt design and later improved to 100% after refining the prompts. However, the accuracy rate may decrease if the framework is tested under different settings.

TABLE 4 Illustration Results on ESG Reports

	GPT-4 - initial test	**GPT-4 - refined prompts**	**Authors**
Total Count of Data Points	90	90	90
Actual Count of Correct Data Points	84	89	90
% Correct Data Extraction	93.3%	98.9%	100%
Total Time to Extract Data (in minutes)	5	2.5	45
PDF Conversion Time	2.5	NA	NA
Code Running Time	2.5	2.5	NA

Note: This table shows the performance evaluation results of the framework for financial data extraction tasks on ESG reports. A total of 90 data points were collected from 15 HKEX-listed companies' ESG reports. The reports were randomly selected from the years 2021 to 2023. The accuracy rate reached 93.3% with the initial prompt design and later improved to 98.9% after refining the prompts. However, the accuracy rate may decrease if the framework is tested under different settings.

APPENDIX 1 Variable List for the Illustration on ACFRs

	Variables	Statement	Account Name
1	Assessed Value	Assessed Value of Taxable Property	Fiscal Year Ended June 30, 2022, Assessed Value Total
2	General Fund Unassigned	Balance Sheet Governmental Funds	Fund Balances Unassigned, General
3	General Fund Assigned	Balance Sheet Governmental Funds	Fund Balances Assigned, General
4	Total Assets	Balance Sheet Governmental Funds	Assets - Total Assets, Total
5	Cash and Investments	Balance Sheet Governmental Funds	Assets: (Cash and Investments; Restricted Cash and Investments), General
6	Long-Term Liabilities for Governmental Activities	Note 10. Long-Term Obligations	Governmental Activities - Total Governmental Activities, Balance June 30, 2022
7	Program Revenues	Statement of Activities	Total Primary Government, Program Revenues (Charges for Service; Operating Grants and Contributions; Capital Grants and Contributions)
8	General Revenues and Transfers	Statement of Activities	Total General Revenues and Transfers, Net (Expenses) Revenues and Changes in Net Position Total
9	Expenses	Statement of Activities	Total Primary Government, Expenses
10	Operating Grants and Contribution	Statement of Activities	Total Primary Government, Program Revenues Operating Grants and Contributions
11	Capital Grants and Contributions	Statement of Activities	Total Primary Government, Program Revenues Capital Grants and Contributions

12	Unrestricted Aid Reported with General Revenue	Statement of Activities	Grants/contributions not restricted, Net (Expenses) Revenues and Changes in Net Position Total
13	Change in Net Position	Statement of Activities	Change in Net Position, Net (Expenses) Revenues and Changes in Net Position Total
14	Unrestricted Net Position	Statement of Net Position	Net Position Unrestricted, Primary Government Total
15	Long-Term Liabilities	Statement of Net Position	Liabilities - Due in more than one year, Primary Government Total
16	Net Pension Liability	Statement of Net Position	Liabilities - Net Pension Liability, Total
17	OPEB Liabilities	Statement of Net Position	Liabilities - Other post-employment benefits (OPEB), Total
18	GF Expenditures	Statement of Revenues, Expenditures, and Changes in Fund Balances Governmental Funds	Expenditures - Total Expenditures, General
19	Operating Revenues	Statement of Revenues, Expenditures, and Changes in Fund Balances Governmental Funds	Revenues - Total Revenues, General

APPENDIX 2 Example Prompts of the Illustration on ACFRs

Panel A: Examples of *Instruction Learning*:

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

[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 cannot be found, return an empty string ('') for that value.

Rule 4. Only output the JSON format data, with key names as “Expense”, “Charge_for_service”, “Operating_Grant_Contribution”, “Thousand”, and “Million”. Do not output your analytical procedures and explanations.

[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. If the values in the table are expressed in thousands, output 1000. Otherwise, output an empty string ('').
5. If the values in the table are expressed in millions, output 1000000. Otherwise, output an empty string ('').

Panel B: Example of *few-shot learning*:

[Task]: 1. Long-Term Liabilities for total activities:

a. Usually, they are all the line items between “Long term liabilities/noncurrent liabilities” and “total liabilities”/“total noncurrent liability”/“total liabilities due in more than one year”.

b. Extract only the values for total activities and do not miss any line item.

c. Some example names for the line items include: “Lease liability”, “Compensated absences payable”, and “Post-closure care costs”.

Panel C: Example of *Chain-of-Thought Prompting*:

[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*] format.

APPENDIX 3 Pseudocode for the Batch Querying Function

```
    Function batch_querying 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
      4. Retrieve the first choice's message content from the
         completion response
      5. Return the content of the first choice
End Function
```

APPENDIX 4 Variable List for the Illustration on ESG Reports

	Variables
1	The amount of Air emission NOx in grams
2	The amount of Air emission SOx in grams
3	The amount of direct greenhouse gases (Scope 1) in tonnes
4	The amount of energy indirect greenhouse gases (Scope 2) in tonnes
5	The amount of total hazardous waste produced in tonnes
6	The amount of total non-hazardous waste produced in tonnes