Accounting and Management Information Systems
Vol. 25, No. 1, pp. 70-105, 2026
DOI: http://dx.doi.org/10.24818/jamis.2026.01003

Prompting to Extract Data Inputs for Accounting Systems from Heterogeneous Data Sources

Florina G. Hutter[a], Juliane Wutzler[1,b]

[a]*Hasso Plattner Institute, Germany*
[b]*Worms University of Applied Sciences, Germany*

Abstract

Research Question: Can Large Language Models (LLMs) be used as low-cost tools to efficiently and effectively extract data from heterogeneous sources fed into accounting systems and processes?

Motivation: Accounting departments use a variety of data from a wide range of sources and feed them as inputs into their accounting systems and processes. Extracting such data often requires manual effort. Large Language Models may be a low-cost way to extract data without substantial upfront investments. Prior literature documents the potential of LLMs for data extraction in other domains or for long and largely semantic accounting documents. While these guidelines may be transferable to semantic data feeding into accounting systems, such as order emails, they are likely not directly transferable to non-semantic, semi-structured data sources, such as invoices.

Idea: In our proof-of-concept, we test whether general prompting guidelines from prior literature apply to both non-semantic but semi-structured (i.e., invoices) and semantic but unstructured data sources (i.e., order emails) used as inputs into the accounting system. We then identify these issues and derive guidelines for practical use.

Data: A synthetic dataset consisting of 46 heterogeneous PDF invoices and 10 order emails in Outlook format was created. Synthetic data allow explicitly including challenging variations and eliminate data privacy concerns.

Tools: Following design science research, we test and improve LLM (Mixtral-8x7B) prompts for Named Entity Recognition derived from prior literature to establish accounting-specific prompt guidelines.

---

[1] *Corresponding author:* Area of Tourism and Transportation, Worms University of Applied Sciences, Erenburgerstraße 19, 67549 Worms, Germany, Tel: +49 6241 509118, email addresses: wutzler@hs-worms.de.

Funding: there is no funding for this research.

Article History: Received December 20, 2025; Accepted February 27, 2026.
Accepted by: Sînziana-Maria Rîndașu.

Findings: Using Large Language Models to extract data that can be used as inputs into accounting processes requires case-specific adjustments to general prompting guidelines derived from the literature. We develop solutions to problems resulting from general prompting guidance and define transferable strategies for creating prompts that allow the extraction of data from semantic and non-semantic accounting data sources.
Contribution: We provide guidance on how LLMs can extract data for use in accounting systems. Data sources differ substantially from those in prior literature.

Keywords: generative artificial intelligence, large language models, prompting, data extraction, accounting systems

JEL codes: M49

1. Introduction

Accountants often spend considerable time extracting data from structured and unstructured sources. While major advances in Document Artificial Intelligence (AI) have been observed over the last decades (Cui *et al.*, 2021), existing data extraction methods often encounter substantial challenges, making them costly, time-consuming, and often error-prone. First, they generally rely on model training, which requires substantial time investments, e.g., for labelling data or validating extraction decisions and providing feedback for learning (Klein *et al.*, 2004). Optical Character Recognition (OCR) in combination with other technologies such as regular expressions or pattern matching, is often used to extract individual pieces of information from PDF documents. Invoices are challenging candidates for OCR as their graphical layouts may differ substantially on a case-by-case basis (Ha & Horák, 2022). When, for example, applying regular expressions to invoice data, rules need to be consistently monitored and adjusted. Traditional rule-based pattern-matching approaches also fail when input texts become heterogeneous (Liu, 2025), which is often the case in the accounting context, especially when it comes to invoices. In addition, while commercial tools such as ABBYY Fine Reader, DataSnipper, and OCRMiner show good, but far from perfect, levels of accuracy, their implementation and integration require large resource commitments, both financially and with respect to human resources by IT staff and accounting experts.

With the introduction of Large Language Models (LLMs), a new and potentially efficient tool for data extraction may be available (Huang *et al.*, 2024). While this methodology is already being explored in other disciplines such as medicine, the extraction of clinical data is fundamentally different from data in the accounting domain as they are (1) often already delivered in electronic format and (2) mostly in unstructured narrative (i.e., semantic) form. So far, little research has focused on the

application of LLMs to the extraction of accounting data. The focus is increasingly placed on the extraction of large-scale financial information from annual reports and sustainability reports. These are unique types of documents, as they are usually long documents with large text-based components, as well as a smaller proportion of tabular data. Findings from these studies are mostly interesting for the auditing and consulting community which may want to scan such documents or for researchers tapping new data sources. Many of the studies in this area focus on the process of data extraction but remain agnostic as to how exactly to prompt the LLM to obtain a high-quality output. So far, research has, however, neglected one specific group of accountants who have a particular need to extract data on a daily basis: accounting functions within businesses.

Our study fills this gap by adjusting and extending existing general prompting guidelines to the context of accounting data used in the business context. Specifically, we capture the heterogeneity of this task by creating two synthetic datasets with one containing unstructured but semantic data, which is expected to be a natural candidate for LLM processing, and the other containing (semi-)structured but non-semantic data from invoices. To do so, we apply a literature-based prompt to our sample and analyse extraction issues. In an iterative manner, we optimise this prompt to solve the identified challenges using prompting. Ultimately, we codify these insights in the form of best-practice prompts and prompt guidelines (Moundas *et al.*, 2024) for users to apply to identical use cases (i.e., invoices and order emails) or similar tabular and semantic data extraction tasks in the accounting domain.

We thereby extend prior literature in three ways. First, we take the overall process of data extraction from PDF to individual pieces of information (e.g., Li *et al.*, 2025) as given. Instead, our proof-of-concept focuses on one part of this extraction process, namely, the interaction with the LLM, and shows how to optimise data extraction via prompting. Second, while prior studies have focused on extracting a larger amount of information related to a general topic or identifying the relevant pages and sections in very long documents, we focus on short documents of often only one page in length, which contain a variety of distinct pieces of information to be individually extracted. Third, Li *et al.* (2025) and similar studies focus on extracting large amounts of data from annual reports with the goal of using this information in a research setting – not to be entered into ERP systems and thereby becoming critical for business operations. While for Li et al.'s approach, errors in data collection do not matter if they are unsystematic and the law of large numbers applies, inputs to accounting systems need to have high levels of accuracy as misstating transaction data can easily lead to malfunctioning business operations and issues during the audit process.

2. Literature review

2.1 Automation and generative AI in the accounting function

As it becomes more difficult to further increase productivity in operational areas, businesses increasingly focus on enhancing efficiency in administrative areas such as accounting and finance departments to reduce costs. Additionally, an increasing workload, for example, due to the implementation of new and complex accounting standards (e.g., IFRS 16) or additional tasks (e.g., sustainability accounting and reporting) is observed. The automation of standardised routine accounting tasks offers a potential solution to this challenge.

Initially, automation within the accounting and finance function was done, at most, for basic repetitive tasks within MS Excel worksheets using macro programming or by customising ERP systems. In both cases, the tasks can only be automated within a single system. With the introduction of robotic process automation (RPA), which emulates human actions on screen and thereby allows working across different systems (Cooper *et al.*, 2019; Lacity & Willcocks, 2016; Willcocks *et al.*, 2015), the scope for automation has been extended. However, both research and practice soon became aware of its limitations such as high failure rates (Moffitt *et al.*, 2018) and governance issues (Bakarich & O'Brien, 2021). Ultimately, these limitations lead to efficiency gains from RPA that are much lower than expected. In the next wave, RPA bots were combined with other powerful technologies, such as optical character recognition (OCR) and AI, for more complex information processing.

More recently, Generative AI (GenAI) in general and Large Language Models (LLMs) in particular, have disrupted the way in which administrative tasks are performed. The earliest use cases were subject-agnostic and included summarising email content or meetings to obtain quicker access to information. LLMs were quickly able to solve accounting tasks at least at a satisfactory level. Eulerich *et al.* (2024) show that while performance of ChatGPT 3.5 was weak, ChatGPT 4 was able to pass all sections of major accounting certification exams (i.e., CPA, CMA, CIA, and EA). In fact, they provide evidence that the LLM performs at least as well as accounting professionals do. Similarly, Cheng *et al.* (2024) analyse the ability of ChatGPT models 3.5 and 4 to complete accounting tasks by using them to solve educational accounting cases. Their analysis shows that LLMs perform much better when it comes to purely semantic tasks (e.g., explaining or applying rules) than in the non-semantic context, when, for example, preparing financial statements.

Early use cases focused on using LLMs for purely text-based tasks such as idea generation in (internal) audits (e.g., risk identification, audit scope, interview preparation and summarisation (Emett *et al.*, 2023)), summary tasks (e.g., interview summaries in internal audit (Eulerich & Wood, 2023)), or explanatory tasks (e.g.,

regulation and governance (Cheng *et al.*, 2024)). As existing text mining methods are often ineffective at obtaining granular data (Bochkay *et al.*, 2023; Senave *et al.*, 2023), the use of LLMs for data extraction has attracted attention. Specifically, prior literature has documented how LLMs can help extract data from long unstructured but largely semantic accounting documents such as annual reports and Economic, Social, and Governance (ESG) reports. Huang *et al.* (2023) developed an LLM known as FinBERT, specifically designed to summarise contextual information in financial texts. This tool significantly outperforms other established data extraction approaches (e.g., naïve Bayes, support vector machine). It also outperforms other models with respect to extracting data on specific topics such as ESG passages.

Based on the principles of design science research, Li *et al.* (2025) present a framework for extracting text from PDF documents, thereby providing researchers with access to large amounts of previously unused unstructured data. They show that their extraction framework achieves high accuracy rates of up to 100% for governmental Annual Comprehensive Financial Reports (ACFRs) and up to 98.9% for ESG reports. Out-of-sample tests also show high levels of accuracy of 96%. Föhr *et al.* (2023) focus on the prompting component and provide a framework for sustainability-related audit prompting with the goal of validating and verifying their information. Ni *et al.* (2023) take a different approach. Instead of providing a framework to enable researchers to extract data themselves, they propose a ready-made system (CHATREPORT), which can be used to analyse sustainability reports along a variety of dimensions. Specifically, the system creates a score and allows conducting an interactive analysis.

However, hallucinations, which reflect „generated content that is either nonsensical or unfaithful to the provided source content" (Huang *et al.*, 2024, p. 2), remain an important issue (Huang *et al.*, 2024). When extracting data for further analysis, this issue can be addressed by making the LLM output traceable (Ni *et al.*, 2023). When using LLMs to extract large amounts of data from annual reports for large sample-based research, an occasional hallucination issue is likely to be irrelevant for hypothesis testing as long as it is unsystematic and the law of large numbers applies. This shows that the literature has investigated large-scale data extraction, mostly for analysis or research purposes, as outlined above. The use of LLMs to obtain and standardise transactional data to be fed into accounting systems has not yet been explored. While much accounting information flows into accounting systems through integrated solutions (e.g., production data on raw material usage), accountants frequently receive unstructured documents containing transaction data that need to be transferred into journal entries. So far, there is no standardised state-of-the-art technology and process used to approach the problem of data extraction for the purpose of feeding data into ERP systems.

We expect the application of LLMs for extracting data from annual reports or sustainability reports for research or analysis purposes to differ from the application for extracting information to be fed into accounting systems (e.g., from invoices or orders) for the following reasons:

1) Annual reports are well-structured (e.g., standardised recurring headings, consistent formatting) long-form documents with large quantities of semantic texts that are supported by non-semantic information in table format. In contrast, invoices are characterised by containing almost exclusively non-semantic data in heterogeneous table format (e.g., different headers, units of measurement, date formats). The latter is optimised for human readability and not for machine-reading and parsing.
2) When extracting large amounts of data for research purposes, minor errors can be neglected if these are unsystematic. However, accuracy is crucial for accounting inputs, as accounting must be free from error (IFRS Conceptual Framework). In the case of material errors, restatements with costly consequences, such as enforcement actions, negative restatement announcement returns, or higher executive turnover may be required (e.g., Peterson, 2012). Hence, requirements in terms of accuracy (i.e., error rates and hallucinations) are much higher for accounting system inputs.

The key to achieving an optimal output is to use optimised prompts when interacting with an LLM. However, what such optimal prompts look like in terms of prompt components, structure, wording, etc., differs strongly depending on the use case and data sources. Thus, we build on prior work that defines frameworks for information extraction in accounting (e.g., Li *et al.*, 2025) and extend it by defining the following two research questions:

1) *Do general prompting guidelines apply to data extraction in the accounting and finance function?*
2) *What constitutes good prompting for structured versus unstructured data used in the accounting context based on prior literature?*

2.2 Literature-based prompting guidelines

LLMs produce text based on learned linguistic patterns (Ida, 2024). The underlying transformer architecture predicts the probability of the next token in a given sequence. It sequentially generates an output sequence according to these probabilities by appending the chosen token to the existing token sequence. The LLM repeats this probability calculation until the output is complete. Through this mechanism, LLMs effectively perform core natural language processing (NLP) tasks such as text generation and completion (Paaß & Giesselbach, 2023). The output is triggered by a prompt (sequence) containing a set of instructions provided to the LLM. This input token sequence determines the probability that the next token will be selected (Paaß & Giesselbach, 2023; White *et al.*, 2023).

Table 1. ABC adoption measurements

	GUIDELINE	SUPPORTING LITERATURE
1	**Set the fundamentals** - Choose a suitable *LLM* and user *interface* - Set *inference parameters* - *Understand* LLM's general response *behaviour*	Dang *et al.* (2022); Ekin (2023); Liu and Chilton (2022); Liu *et al.* (2023); Lu *et al.* (2021); Wu *et al.* (2022); Zamfirescu-Pereira *et al.* (2023)
2	**Specify the desired output** - Define a *ground truth* - Define r*ules, principles, and constraints*	Ekin (2023); Hu *et al.* (2024); Schmidt *et al.* (2024); White *et al.* (2023)
3	**Keep it short, clear, and simple** - Use a *clear and concise writing style* - Use *positive phrasing* and *neutral tone* - Optimize *trigger-keywords* and sentences - *Reduce complexity* (e.g. by multi-prompt learning or Chain-of-Thought prompting)	Dang *et al.* (2022); Ekin (2023); Hu *et al.* (2024); Kojima *et al.* (2022); Korzyński *et al.* (2023); Li *et al.* (2025); Liu and Chilton (2022); Shin *et al.* (2020); Wang *et al.* (2023); Wu *et al.* (2022); Wutzler (2023); Zamfirescu-Pereira *et al.* (2023)
4	**Communicate groundwork for understanding** - Provide *context* and *background* information - Define *notation and terminology* - Consider *edge cases*	Dang *et al.* (2022); Ekin (2023); Moundas *et al.* (2024); Schmidt *et al.* (2024); White *et al.* (2023); Wu *et al.* (2022);
5	**Structure the prompt** - Use optimal *template* (Table 2) - Use *guidance sentences* - Assign a *role* (e.g., expert, teacher) - Specify a *task* (e.g., extract, list) - Describe the *output preferences* (Guideline 2) - Provide *examples* - *Repeat* important aspects	Akcali *et al.* (2025); Dang *et al.* (2022); Ekin (2023); Hu *et al.* (2024); Kirstain *et al.* (2021); Kojima *et al.* (2022); Korzyński *et al.* (2023); Li *et al.* (2024); Li *et al.* (2025); Liu *et al.* (2021); Liu *et al.* (2023); Mistral AI (2025); Moundas *et al.* (2024); Polat *et al.* (2024); Ray (2023); Wei *et al.* (2022); White *et al.* (2023); Zamfirescu-Pereira *et al.* (2023); Zhao *et al.* (2021)
6	**Define data extraction specifications** - Specify and describe the named entities for Named Entity Recognition (NER)	Hu *et al.* (2024); Li *et al.* (2025); Moundas *et al.* (2024)
7	**Iterative prompting** - *Adjust*: Try variations of Guidelines 1-6 application - *Interact* with the LLM for trouble shooting - Testing using *heterogeneous and representative datasets*	Dang *et al.* (2022); Ekin (2023); Korzyński et al. (2023); Moundas *et al.* (2024); Ray (2023); Schmidt *et al.* (2024); White *et al.* (2023); Wu *et al.* (2022); Zamfirescu-Pereira *et al.* (2023)
8	**Human oversight** - Need of *subject matter experts*	Hu *et al.* (2024); Liu *et al.* (2023); Ray (2023)

To elicit high-quality LLM output, prompts should be well structured and precisely formulated, as even minor prompt changes may alter the output (Liu *et al.*, 2023; White *et al.*, 2023). The process of optimising prompts is known as "prompt engineering" (Knoth *et al.*, 2024). It is often characterised by extensive trial and error, iterative testing, and the evaluation of different prompting strategies.

Specifically, wordings, punctuation, instructions, and additional information in the prompt are changed progressively. Output differences are documented after each step. Subsequently, possible combinations of these changes are tested individually until the LLM output yields sufficiently precise results. The literature identifies different characteristics of high-quality prompts. Table 1 summarises prior findings in the form of guidelines.

1) *Set the fundamentals*

LLM providers often offer user *interfaces* that make LLM interactions more user friendly. In addition, most LLMs can be accessed via an Application Programming Interface (API).

Output quality highly depends on the choice of the underlying LLM and its pretraining (Paaß & Giesselbach, 2023). While some models perform better when handling structured content, others are designed to be talkative and produce creative text. Furthermore, LLMs differ in their size. The larger a model, the more likely it is to support multiple tasks. However, this comes at the cost of reduced precision. Larger models are not necessarily a better choice as they have a higher probability of hallucination and are costlier to operate in terms of computing power (Lipenkova, 2022). Another determinant for choosing the most suitable LLM is the context window from which the models can assimilate information. It is the maximum number of tokens that an LLM can work on without "forgetting" what has previously been discussed (Ruiz *et al.,* 2024). A larger context window allows a more detailed description of tasks and may yield better results. Still, the number of tokens used for the instruction determines the cost of LLM usage and should be kept to a minimum (Bergmann, 2024).

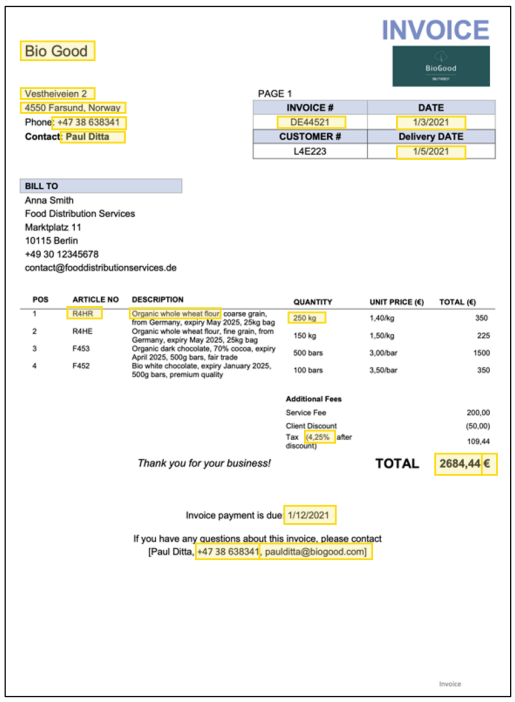
In addition to textual prompt input, most user interfaces and LLM APIs provide the option of setting *inference parameters* to control how the model should respond to a prompt. Settings that can be specified include but are not limited to (1) the minimum and maximum number of tokens used for the output generation, (2) setting a "stop sequence" which defines a sequence of characters after which the model will stop generating output, and (3) "greedy" and "sampling" decoding methods in NLP. These factors define how conservative or creative the LLM is. Greedy decoding takes the highest probability tokens and leaves less room for creativity. Sampling decoding allows for more creativity and can lead to more differences in outputs for the same input. However, this variability increases the risk of nonsense (Ekin, 2023; Paaß & Giesselbach, 2023).

Finally, when using a new LLM, users should familiarise themselves with the general *behaviour* of the LLM (e.g., when creating a table, completing a sentence) to identify and address potential issues early (Liu & Chilton, 2022).

2) *Specify the desired output*

To ensure that the output aligns with user expectations, users should specify a *ground truth* that defines an ideal output, and thereby the writing style (e.g., friendly or formal), language, data format, length, structure, and complexity (see Figure 1 for example). Once established, these requirements can be used to determine *rules, principles, and constraints* for the LLM to adhere to.

Input: Invoice (PDF)



Output: Ground Truth (JSON)


```
{
    "InvoiceNo": "DE44521",
    "InvoiceDate": "03.01.2021",
    "ExpectedDeliveryDate": "05.01.2021",
    "SupCompany": "Bio Good",
    "SupContactPerson": "Paul Ditta",
    "SupEmail": "paulditta@biogood.com",
    "SupPhone": "+47 38 638341",
    "SupAddress": "Vestheiveien 2",
    "SupCity": "Farsund",
    "SupPostalCode": "4550",
    "SupCountry": "Norway",
    "ProList": [
        {
        "ArticleNo": "R4HR",
        "Name": "Organic whole wheat flour",
        "Quantity": 250,
        "QuantityUnit": "kg"
        },
        {
        "ArticleNo": "R4HE",
        "Name": "Organic whole wheat flour",
        "Quantity": 150,
        "QuantityUnit": "kg"
        },
        {
        "ArticleNo": "F453",
        "Name": "Organic dark chocolate",
        "Quantity": 500,
        "QuantityUnit": "bars"
        },
        {
        "ArticleNo": "F452",
        "Name": "Bio white chocolate",
        "Quantity": 100,
        "QuantityUnit": "bars"
        }
    ],
    "TaxRate": "0,0425",
    "TotalPrice": "2684,44",
    "CurrencySymbol": "€",
    "DueDate": "12.01.2021"
}
```


Figure 1. Sample invoice in PDF format versus ground truth in JSON format

3) *Keep it short, clear, and simple*

In general, a *clear and concise writing style* helps eliminate ambiguity and minimises misunderstandings, errors, and omissions (Li *et al.,* 2025). Therefore, users should use simple and *positive phrasing* (e.g., "do" instead of "do not") while maintaining a *neutral tone* and avoiding unnecessary politeness (Wutzler, 2023; Zamfirescu-Pereira *et al.,* 2023).

Furthermore, LLM output has been shown to be strongly impacted by certain *trigger-keywords*. These may be general (e.g., “remember”; list”, “name”, “explain”, “concise”; “comprehensive”; “strictly”) or domain-specific in nature. Users should focus on identifying use-case-specific keywords during the iterative prompting process and strategically employ both general and specific ones (Shin *et al.,* 2020).

In addition, LLMs have been shown to benefit from *reducing* task *complexity* by providing step-by-step instructions or by using a multi-prompt approach. This involves using the output of one prompt as the input for the next prompt. Additionally, Chain-of-Thought prompting should be applied, where the LLM is instructed to explain its reasoning step-by-step. This approach is particularly beneficial for mathematical and logical tasks (Kojima *et al.,* 2022; Korzyński *et al.,* 2023; Wu *et al.,* 2022).

4) *Communicate groundwork for understanding*

Prompts should provide the LLM with relevant *context* and *background* information, such as explaining the general task, intent, scope, role, and document types. However, users must balance too little and too much additional information and instructions (see Guideline 3), as additional information may increase the risk of misunderstanding and thus generate suboptimal output (Dang *et al.,* 2022; Wu *et al.*, 2022).

Furthermore, at the beginning of the prompt, users should define *notation* and *terminology* (e.g., "*" to indicate "new"; domain-specific terminology) (White *et al.,* 2023).

For greater accuracy, *edge cases* should be addressed within the prompt. That is, users should define how the LLM should behave in unexpected situations (e.g., define what the LLM should output if the answer is unknown or uncertain) (Stryker & Scapicchio, 2024).

5) *Structure the prompt*

Once users have identified all relevant prompt components, they should focus on optimising the structure of the prompt, which has been shown to strongly influence the generated response. Usually, model developers suggest that specific *templates* that follow a logical order should be used for the prompt to deliver the best result. These can involve indentations, parentheses, or keywords (Mistral AI, 2025). Table 2 provides examples of prompt templates.

When deviating from prompt templates, users should rely on *guidance sentences*/control phrases that clearly indicate what the following section of the instructions relates to (e.g., "Follow these guiding principles:"; "Use the following example as reference for your output."; "Input", "Output"). In addition, users should include a placeholder for the LLM output (ending cues) (Korzyński *et al.*, 2023). In general, users should begin by assigning a *role* to the LLM to ensure that it uses specific context-related expertise or assumes certain responsibilities. Next, users should pose a question or *task* with detailed instructions that contain descriptions of *output preferences* (e.g., length or structure; see Guideline 2). This approach provides the LLM with clear guidance regarding the desired format and structure of the output (Ray, 2023). Prompts allow for in-context learning where LLMs learn

from demonstrations (instructions and examples) in the prompt rather than from knowledge stored in the model weights during training time (Gemini Team Google, 2024). This can be used instead of the more resource-intensive approach of finetuning, where weights and parameters of a model's artificial neural network are changed using a manually labelled, subject-specific dataset (e.g., financial data). When using the less resource-intensive in-context learning, users provide sample inputs and outputs. While prompts can be created without *examples* (i.e., "zero-shot" prompting), providing a single (i.e., "one-shot" prompting) or multiple (i.e., "few-shot" prompting) input/output examples have been shown to lead to superior output (Dang *et al.,* 2022). Depending on the application area, the number of examples that positively affect output can differ (Kirstain *et al.,* 2021). Examples should be placed towards the end of the prompt.

Repetition also increases output quality, although it contradicts the aim of minimising prompt length and thereby the prompt's token number (Zamfirescu-Pereira *et al.,* 2023; see Guideline 3). Again, it is important to find a balance between too much and too little information (see Guideline 3).

Table 2. Sample Prompt Structure Templates

#	Template	Reference
1	LLM provider suggestions: <s> [INST] *[instruction]* [/INST]	Mistral AI (2025)*
2	Structure: (1) Role; (2) Task; (3) Format, Principles, Rules; (4) List of entities; (5) Examples; (6) Reminder and Output Guidance	*
3	Q: *[X]*. A: *[T]*	Kojima *et al.* (2022)
4	Given the *[input]*, *[action verb] (e.g., organise) [data types] (e.g., list)* of *[output]*, *[few shot examples]*	Wu *et al.* (2022)
5	Extractor pattern: Extract *[generation_contraints]* in the format *[extraction_pattern]* from *[input_specification]*	Moundas *et al.* (2024)

Notes: * indicates prompt structures used in the study at hand's analysis

6) *Define data extraction specifications*

NLP enables a wide range of tasks, including document retrieval, code generation, text generation, summarisation, classification, clustering, sentiment analysis, translation, question answering, retrieval-augmented generation, dialog systems, and data extraction. Each of these tasks may require slightly different prompting techniques (Ekin, 2023; Paaß & Giesselbach, 2023). When using LLMs to extract data, Named Entity Recognition (NER) is a suitable technique. In this case, the prompt should explicitly specify the entities to be extracted and provide detailed

descriptions to ensure that the LLM can accurately identify them in the input document (Moundas *et al.,* 2024).

7) *Iterative prompting*
Once a baseline prompt has been created, users should analyse the output, identify potential errors, and *adjust* the prompt (see Guidelines 1-6) in an iterative manner (Ray, 2023). As part of this iterative process, users may opt for *interacting* with the LLM by asking questions to identify reasons why an output was generated in a particular manner and how the prompt could be adjusted to improve the output (White *et al.,* 2023). To verify the robustness of the prompt across varying input sequences, users should evaluate it using multiple *heterogeneous and representative datasets* (Moundas *et al.,* 2024).

8) *Human oversight*
Ultimately, output quality may vary even for repeated usage of an identical prompt. Therefore, human oversight from *subject-matter experts* is essential to mitigate the common risks associated with LLMs, including hallucinations, biases, and inconsistent outputs (Hu *et al.,* 2024; Ray, 2023; Stryker & Scapicchio, 2024; Zewe, 2023). As the transferability of prompts is poor whenever the data structure or output requirements change, a new prompt must be developed or an existing prompt customised for each task (e.g., when extracting information from invoices versus emails) (Liu *et al.,* 2023).

3. Research design

3.1 Methodological approach

We follow design science research (DSR) (Gregor & Hevner, 2013) to design and validate guidelines for extracting data from accounting-related documents using LLMs. Data sources that provide inputs to accounting processes are extremely heterogeneous, ranging from standardised structured formats (e.g., e-invoices; order forms) to customised but structured formats (e.g., individualised invoices) to fully unstructured formats (e.g., order emails). However, as at the same time data extracted from such heterogeneous sources must be free from error (IFRS Conceptual Framework), guidelines for data extraction using LLMs identified in prior literature (see Table 1) may not be readily transferable to this context. In line with DSR, we apply design guidelines derived from prior literature to accounting input data, with the goal of writing entities into a JSON format file. We choose this format because it is one of the most popular formats for exchanging data, it is human readable and seamlessly integrates with most relational database systems (Petković, 2017). We require the LLM to generate the output using NER, where a named entity can be a word or phrase identified by a specific name, expressing for instance, the name of an organisation, person, and geographic location, but it can also be a currency, time,

or percentage (Li *et al.,* 2020). The NER function proves useful in (financial) document parsing, as it allows the extraction of structured data from unstructured financial texts. To verify whether the JSON output is satisfactory, we establish a ground truth for each invoice. By manually comparing LLM outputs against the ground truth, in line with Bose *et al.* (2021), we assess the reliability of the results by discovering and correcting irregularities and mismatches that were generated by the LLM to ensure high-quality data output. More precisely, we check whether the extracted entity values match the ground truth. If this is not the case, the entity is defined as incorrect. The number of correct entity extractions is then considered relative to the total number of entities to determine the error rate.

We select the Mixtral-8x7B model by Mistral AI because it has a comparably large context window of approximately 32,000 tokens, can handle multiple languages, and performs well with respect to structural activities. It is suitable for financial contexts, in which problems are often multifaceted and domain-specific. The model leverages a collection of expert models, each tailored to address and solve a distinct component of the problem, which delivers more accurate and context-aware outputs (Mistral AI team, 2023). The Mixtral-8x7B model was accessed via an API providing freedom with respect to the prompt structure and parameter settings. In the business context, the use of an API has two distinct advantages. First, the modular nature of accessing the LLM through an API adds flexibility to modify or even remove the step from the invoice processing operation. Second, processing via an API can be automated and, thus, easily scaled up (Li & Vasarhelyi, 2024).

We then identified issues that arise when applying a literature-based prompt to semantic (i.e., order emails) and non-semantic (i.e., invoices) data sources. In line with DSR, we follow an iterative process and adjust the guidelines to reflect requirements specific to extracting data from heterogeneous sources as inputs for accounting processes.

3.2 Data

Transactional accounting data entering accounting systems may be received in the form of structured or unstructured semantic, semi-semantic, or non-semantic documents. To capture input heterogeneity, we create two synthetic datasets: (1) semi-structured but non-semantic invoice data and (2) unstructured but semantic order email data for a fictional food retailer. Synthetic data are increasingly chosen to protect companies' sensitive private financial data (Beduschi, 2024). Basing invoice creation on publicly available real-world invoice examples allows specifying variations in layout, formats, and writing that potentially cause difficulties in data extraction.

3.2.1 Invoice dataset (semi-structured, non-semantic)

As invoices are non-semantic documents that contain only a few word sequences in natural language, they are characterised by a semi-structured mix of words and numbers. Data are placed in different locations across the document depending on the context. For example, contact information is generally found in the header or footer, whereas order line items are displayed in the tabular body of invoices. Sometimes, the same information can be included several times (e.g., contact name in the header and footer) (Hamdi *et al.,* 2021; Saout *et al.,* 2024). Our invoice dataset comprises 46 invoices that can be divided into two subsets. The first subset consists of 25 PDF documents from 25 different suppliers, all of which use different templates and display different contents. 20 documents are written in English while five documents are written in German to capture language barriers that international businesses may face. The second subset contains 21 PDF documents divided into three subsets of seven PDF documents each. One subset of seven invoices uses the same invoice data from the same English-speaking supplier but formatted in six different ways plus a German version of the sixth template. Figure 2 shows an overview of all invoice datasets used. We test for external validity by applying the final prompt to the second subset, which again contains different variations in terms of invoice layout, language (English vs. German), and data content. This addresses potential concerns of introducing too little noise into our synthetic invoice data or overfitting our model prompt to the original dataset.

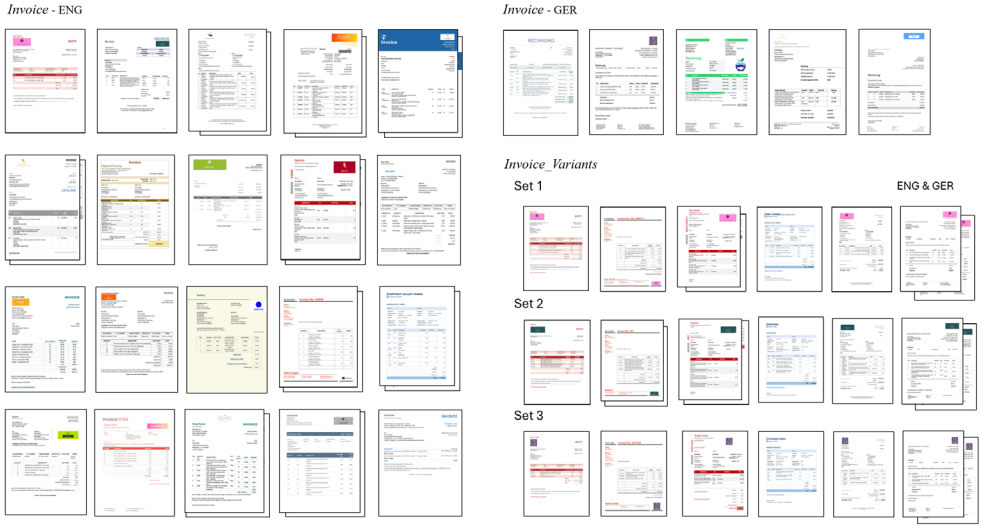


Figure 2. Invoice datasets: an overview

The documents contain the suppliers' contact details, which include varying address or phone number formats. Additionally, differences in writing style (e.g., abbreviations or synonyms) were introduced for the same entity (e.g., "invoice-date" versus "issue-date" versus "document-date"). Other invoice characteristics are invoice IDs or order item IDs, which may vary in length and can be numeric or alphanumeric. The date formats and the number and order of the table columns also vary. The listed order items may or may not have detailed descriptions, depending on the invoice document, and even within a document, unit measurements can differ (e.g., bags, kg/kilos, l/litres). The currency is either €, $, or £. Some information may be written vertically, and some invoices are deliberately missing certain information. Figure 1 displays one sample invoice with its corresponding ground truth.

3.2.2 Email dataset (unstructured, semantic)

Inputs to accounting processes are also frequently unstructured, but semantic in nature. To capture these characteristics, we create a separate dataset with 10 synthetic order emails in PDF format, which in a real-world use case allows for easy access and easy data storage. These share a uniform layout, which is modelled after the Outlook PDF print format of emails, reducing the variance across documents compared with the invoice dataset. Given their semantic nature, the main differences lie in the wording or placement of information such as addresses or ordered items within the email text. Some emails contain special requests, such as a desired delivery date or a request for notifications. Three emails are written in German, whereas the other seven are written in English. Figure 3 displays one sample order email with its corresponding ground truth.

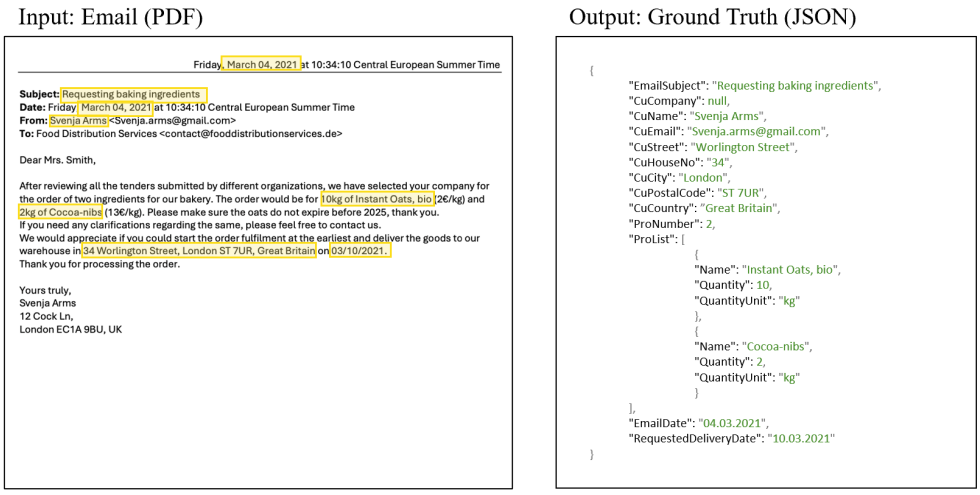


Figure 3. Sample email in PDF format versus ground truth in JSON format

To feed the LLM with the respective data, we follow Li *et al.* (2025) and first convert PDF files into TXT documents using Python. In the second step, we perform the prompting entity extraction task. Finally, in step three, we compare the results of the final prompt with the ground truth (see Figure 4).

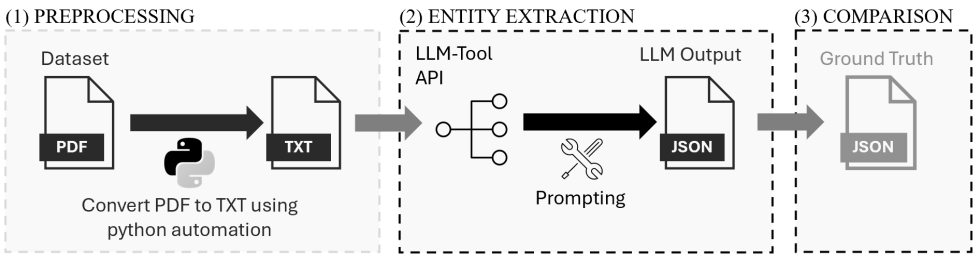


Figure 4. Data extraction process following Li *et al.* (2025)

4. The use of LLMs for accounting data extraction

4.1 Prompt design for semantic and non-semantic data

We apply the prompting guidelines derived from prior literature (see Table 1) to our data. Specifically, we create two prompts: one for unstructured semantic order emails and one for semi-structured non-semantic invoice data. Both follow the structure outlined in Table 2, Template numbers 1 (Mistral specific template) and 2 (general template). We set the inference parameters to "greedy" decoding, allowing for precision, with a minimum token count of "0" and a maximum token count of "800" for the output length. We do not specify a stopping sequence.

For the task at hand, the three most basic instructions are to deliver a JSON output, enter a null value if information is missing, and extract the same expression as written in the input text. The latter instruction allows extracting German words unchanged, if the document is written in German, preventing the LLM from paraphrasing or modifying the data. For the named entity list, we provide telling entity names and include entity-specific instructions that are reinforced by using examples.

4.2 Iteration and evaluation: challenges and solutions

Based on our manual comparison of the output with the ground truth, we identify issues and challenges of the prompt derived from our literature-based guidelines (see Table 1) and iteratively optimise the original prompt to suit the specific requirements of the accounting function (see Table 3 for challenges and solutions).

Table 3. Accounting data specific challenges and solutions

Category	Challenges	Solutions
General Challenges	Multilingualism	Uniform language for prompt examples
	Missing information/entities	Prohibition to enter specific values Reminder to set "null" if not available
	Diverse date formats	General Instruction with date format specific example Reminder to convert
	Multiple dates	No challenge
	Ground truth generation	Merging entities Deleting non-valuable entities Careful consideration of using nested JSON
	Synonyms and abbreviations for entities	Entity-specific explanation
	Confusing supplier and customer data	General Instruction
	Diverse address formats	Nested entities (street, house number) Entity-specific explanation and examples General example
	Different units of quantity	Entity-specific explanation and examples → Errors still occur
	LLM specific issues = Hallucination	Prohibiting instruction at the end Stopping criteria → more prevalent for emails
	Entity Naming	Precise, self-explanatory No underscores in names
	Randomness in output formatting (e.g., phone number)	No prompt solution, but infrequent
Invoice-specific Challenges	Long inference times	No challenge
	Template diversity	Template specific prompt
	Diverse currencies and number formats	General Instruction Entity-specific reminder ("strictly")
	Diverse article number formats	No challenge
	Multiple types of expressions used for one entity (e.g., due date)	Entity-specific format reminder General Instruction to follow in case of uncertainty
	Multiple values given for one entity (e.g., quantity unit)	Template/Supplier tailored prompt
	Unreliable extraction of purely numeric invoice numbers as opposed to alphanumeric ones	No reliable prompt solution
	Random uninstructed calculation (e.g., tax rate)	No reliable prompt solution
Email-specific Challenges	Paraphrasing or omission of unanticipated additional information	No reliable prompt solution

4.2.1 Final optimised prompts

Non-semantic data: The final prompt that delivers the most satisfactory output for the extraction of 16 entities plus four sub-entities from invoices is shown in Figure 5. The aim of this prompt is to be as distinct and short as possible, as the

number of tokens used determines the usage cost, striking a balance between details that deliver high-quality output and details that might confuse the LLM. The final prompt comprises approximately 1,700 tokens. After testing different prompt versions, we conclude that using a one-shot prompt is the best way to receive a satisfactory JSON response, while keeping the token count as low as possible. We carefully select the example invoice based on its complexity, length, and ability to address key challenges.

```
<s> [INST] You are an expert in extracting key information from invoices. Your task is to extract specific entities from a given invoice text that has been converted from PDF files.
Follow these guiding principles when generating responses:
    Return the extracted information as a JSON.
    If an entity is not found in the text, write null.
    Present dates in the following format: DD.MM.YYYY. If the date is 07/04/2022 enter 04.07.2022.
    Present all numbers in the continental European number format: x.xxx,xx.
    Information related to Food Distribution Services is always customer information and never part of the JSON.
    Only extract data related to the supplier.
    Show the exact information given in the input text.
    Special case: If you are not sure what to enter as a value "Input source is unclear.". It is very important that you only use this exact sentence.

The 16 entities to be extracted are listed below:
    - InvoiceNo (holds the text marked as invoice or document number)
    - InvoiceDate
    - ExpectedDeliveryDate
    - SupCompany
    - SupContactPerson (name of the person in charge)
    - SupEmail (If more than one supplier email address is given, list all of them. Keep in mind an email always includes an @ sign.)
    - SupPhone
    - SupAddress (only street and house number)
    - SupCity
    - SupPostalCode (can have different formats, e.g 14169, DE 4558, SP 596)
    - SupCountry
    - ProList (A list of JSONs. Each JSON contains
        - ArticleNo
        - Name (It is important that you only extract the name of the base product. Do not include product descriptions.)
        - Quantity (When something is unclear think logically, 14,4 bags are unrealistic.)
        - QuantityUnit (Various units are possible. Examples are litres, blocks, pieces, kg, bags, packages. If a specification like "5kg unit" is stated, write "5kg unit".)
        of the product.)
    - TaxRate (Never calculate! Holds the percentage as a decimal number. Strictly use a "," as separator. If 15% is given write 0,15. )
    - TotalPrice (Strictly use a "," as separator, for example 4534,45.)
    - CurrencySymbol (for example €, $, £)
    - DueDate (format DD.MM.YYYY)

Use the following example as reference for your output.
###EXAMPLE
Input:
{i9}
Output:
{GT_i9}
###EXAMPLEEND

Extract only the information that is explicitly provided in the input text!
Do not provide any further explanation. Remember to carefully follow the principles mentioned above. Remember to convert the date format properly.
[/INST]
Now generate such a JSON.
Input:
{i3}
Output:
{
```

{i9}: variable holding the input TXT of the example invoice
{GT_i9}: variable holding the JSON ground truth of the example invoice

Figure 5. Optimised prompt for data extraction from invoices (i.e., semi-structured; non-semantic)

Semantic data: While we use the same prompt structure, the prompt for email data extraction, as shown in Figure 6, differs from that used for invoices. A key difference is that instead of a one-shot prompt, we use a few-shot prompt with two examples, as the second example significantly improves output quality. Despite this, the email-prompt contains slightly fewer tokens than the invoice-prompt, with a total of approximately 1,650 tokens, owing to the lower number of extracted entities (i.e., 12 entities) and fewer necessary clarifications and instructions.

The final prompts and issues encountered during the iterative prompt engineering process show that the original literature-based prompting guidelines presented in Table 1 do not allow for dealing with a number of special characteristics

in transaction accounting data. After our evaluation, in line with DSR, we improve upon the original guidelines and indicate adjustments as well as additions (see Table 4).

4.2.2 General Accounting-specific adjustments to prompting guidelines

Data extraction outputs for both semantic and non-semantic data inputs are subject to some systematic issues that require adjustments to the general prompting guidelines. The analysis further shows that the type of data source (i.e., semantic vs. non-semantic) matters substantially for prompt optimisation. Even when extracting similar pieces of information, an identical prompt may not perform equally for both data types:

- **Communicate groundwork for understanding:** Our analysis confirms that even changes in wording (i.e., "you are" vs. "please act as"; blank lines; changes of single characters) alter the context and can, thus, cause significant differences in the output. The issue of changes in wording is particularly prevalent for non-semantic invoice data, as a solution that works well for one invoice often produces errors for another once an adjustment is made – potentially owing to a lack of context in non-semantic data. Semantic data, in contrast, proved to be more sensitive to changes in the examples (i.e., type and structure) provided as part of the input. Thus, context sensitivity must be addressed through iterative testing of every prompt.
- **Structure the prompt:** While the general prompt structure (see Table 2) proved helpful, it does not specify the structure within each category (e.g., *how* to describe the role, *which information* about a task to provide *first*, …). The analysis showed that the location of an instruction within the prompt mattered for output quality. Framing the example using clear keywords in capital letters and key signs (e.g., ###EXAMPLE […] ###EXAMPLEEND) and placing reminders for important requirements introduced by the keyword "remember" towards the end of the prompt improved output quality.

Table 4. Accounting data specific challenges and solutions

	Semantic	Non-semantic
General Accounting-specific Adjustments to Prompting Guidelines		
Communicate groundwork for understanding	Relatively context robust → Iterative prompt testing to identify sensitivities	Highly context sensitive
Structure the prompt – General	Location of information within the prompt matters beyond structure outline in Table 2 → Reminders for general important information towards end of prompt	
– Examples	Few-shot prompting	One-shot prompting
Define data extraction specifications	Errors increase with number of entities to be extracted; order of entities in prompt is relevant as extraction precision decreases for entities at the end of the entity list → Reminders for entity characteristics towards end of prompt	
	Self-explanatory entity names in CamelCase without special characters often sufficient for entity identification. Add explanations only in case of issues	Self-explanatory entity names in CamelCase combined with explanations (i.e., content or pattern definitions) in parentheses required after entity names
	→ Example should cover most peculiarities. To arrive at best example, use iterative trial and error. Additionally, examples can be attached to entities.	
General Accounting-specific Additions to Prompting Guideline		
Multilingualism	Language issues arise from one/few-shot prompting with examples in different languages → Use of one consistent language for prompt and examples regardless of anticipated languages in input	
Formats - Dates	No issues as order email dates standardized in Outlook	Different date formats → Provide explicit instructions, e.g., "convert from 12/01/2025 to 01.12.2025" are superior to "convert from MM/DD/YYYY to "DD.MM.YYYY". Add examples with conversion issue
Formats - Implicit Dates	[no action necessary]	Wrong date calculation if date is explicit (e.g., "due upon receipt") → State what LLM should do when information to be extracted is unclear in general instructions at beginning of prompt
Formats - Numbers	No automatic usage of one consistent number format (e.g., "10.50" vs "10,5") → General instruction at beginning of prompt combined with entity-specific instruction and use of trigger keyword "strictly"	
Formats - Addresses	[no action necessary]	Omitted or wrong house numbers → Extraction together with relevant context (e.g., house numbers with street names)
	[no action necessary]	Omitted or wrong postal codes → Examples within prompt and example input
Multiple outputs/ Hallucination	LLM frequently does not stop generating JSON once all information is extracted → For semantic data, require stopping generating output once JSON is complete	[likely no action necessary; depending on data → refer to semantic data]
Trouble shooting		
Interaction	→ Inquire to LLM why it extracted wrong values; adjust prompt accordingly	
Backwards testing	→ Involves stepwise reversion to prior prompt of superior quality	

- **Define data extraction specifications:** Furthermore, the general prompt structure (see Table 2) remains agnostic regarding the use of NER within the prompt. When the number of entities to be extracted was large, the likelihood of errors increased for entities at the end of the list. Because we placed the due date entity ("DueDate") as one of the last entities to be listed in the JSON, the desired date format was occasionally not used. Hence, in addition to the initial and general date format instruction at the beginning of the prompt, an entity-specific format reminder must be placed at the end of the prompt (see Figure 5, line 33). In addition to the location of entities, their naming is important. Choosing self-explanatory entity names reduces the number of instructions required by the LLM. For example, when extracting the *number* of products ordered by a customer from an order email, naming the respective entity ProNumber instead of ProQuantity or instead of explaining it by "number of different types of products requested" proved superior due to its clear and concise nature. Furthermore, entity names should prefer Camel Case writing (e.g., "SupEmail") instead of including special characters (e.g., "Sup_Email") or spaces (e.g., "Sup Email").

For non-semantic data, the LLM frequently extracted incorrect values, wrote correct values in the wrong format, or produced a combination thereof. Specifically, the address, due date, quantity and quantity unit, and tax rate entities were more likely to cause problems than others. Adding entity-specific explanations in parentheses after the entity name (e.g., see Figure 5, lines 13-33) or a change in the general instruction (e.g., see Figure 5, lines 2-10) frequently solved these issues.

The non-semantic invoice data displayed more challenges than the semantic data. An iterative trial-and-error process was required to identify an instruction and/or naming convention that worked for all invoices. For example, to identify a contact person on the invoice, the entity name "SupContactPerson" was sufficient if the word "contact" was explicitly mentioned in relation to a name on the invoice. For invoices referring to sales staff, the LLM did not provide the name of the contact person. Thus, an additional explanation as "any person in charge" was required (see Figure 5, line 17). Furthermore, email addresses and websites, which both contain ".de" or ".com" endings were confused. Explicitly reminding the LLM to require "@" in email addresses (see Figure 5, line 18), in line with Moundas *et al.*'s (2024) *Pattern Matcher Pattern*, solved this issue. Finally, the risk of confusing customer- and supplier-related data was solved by instructing the LLM that personal data always relates to the supplier and prohibiting to enter data that holds information about the company itself (see Figure 5, line 7).

With respect to semantic data from emails, only few entities required explanations (e.g., "CuName (full name of customer)", see Figure 6, line 12), which is most likely attributed to the additional context, the LLM can draw from. Other entities such as addresses ("CuStreet" or "CuHouseNoCity", see Figure 6, line 14, 15) did not

require explanations at all in the semantic dataset. However, when an order email is sent from an individual person instead of a company, the LLM occasionally enters the name of our own sample company as the value for the ordering company (i.e., CuCompany). An entity-specific reminder to enter null if a private customer order is required (see Figure 6, line 11). This approach aligns with White *et al.*'s (2023) *Refusal Breaker Pattern*, although we apply this method to the issue of not finding a value in the text. Error rates in address extraction could be largely reduced by using few-shot prompting (i.e., two examples).

```
1  <s> [INST] You are an expert for extracting relevant information from emails. Emails contain customer orders for food products.
2  Follow these guiding principles when generating responses:
3      Return the extracted information as a JSON.
4      If an entity is not found in the text, write null.
5      Present dates in the following format: DD.MM.YYYY.
6      Show the exact information given in the input text.
7      Special case: If you are not sure what to enter as a value "Input source is unclear.". It is very important that you only use this exact sentence.
8
9  For each email, extract the following 12 entities:
10     - EmailSubject
11     - CuCompany (company name of customer if a company orders, if not set null)
12     - CuName (full name of customer)
13     - CuEmail (email address from customer)
14     - CuStreet
15     - CuHouseNoCity
16     - CuPostalCode (can have different formats, e.g 14169, DE 4558, SP 596)
17     - CuCountry
18     - ProNumber (number of JSON entries in the "ProList" entity according to the different product types)
19     - ProList (a list of JSONs, only include the following entities, do not include other entities!:
20         - Name (It is important that you only extract the name of the base product. Do not include product descriptions.)
21         - Quantity (integer)
22         - QuantityUnit (the unit  or container in which the quantity of the product is given e.g. litres, blocks, pieces, bars, bags, bottles; even if the text says 100g "unitname" only write
           the"unitname"))
23     - EmailDate
24     - RequestedDeliveryDate
25
26 Use the following examples as reference for your output.
27 ###EXAMPLE
28 Input:
29 {e1}
30 Output:
31 {GT_e1}
32 ###EXAMPLEEND
33 ###EXAMPLE
34 Input:
35 {e2}
36 Output:
37 {GT_e2}
38 ###EXAMPLEEND
39
40 Remember to carefully follow the principles mentioned above.
41 Stop the output after the JSON is completed and all entities are included. Do not provide any further explanation or information. Do not generate a new email.
42 [/INST]
43 Now generate such a JSON.
44 Input:
45 {e4}
46 Output:
47 {
48
```

{e1}: variable holding the input TXT of the example email
{GT_e1}: variable holding the JSON ground truth of the example email

Figure 6. Optimised prompt for data extraction from emails (i.e., unstructured, semantic)

- **Examples:** In line with the literature-based prompting guidance, in-context learning using examples proved to be crucial for high-quality output. However, our analysis shows that semantic and non-semantic data sources require different treatments. When extracting data from our unstructured, semantic email dataset, using multiple examples (specifically, two examples) improved performance. In contrast, for semi-structured, non-semantic invoice data, single-shot prompting (i.e., one example invoice (see Figure 5, line 36)) outperformed few-shot prompting with respect to output quality because of the additional token length. Thus, for invoices, a single well-chosen example that addresses most issues is sufficient. In general, the ideal example is best identified through trial and error. Additionally, specific examples are helpful; for instance, providing some possible values of an entity for guidance (see Figure 5, line 22).

4.2.3 General Accounting-specific additions to prompting guidelines

- **Multilingualism:** Regardless of the prompt language, the LLM works seamlessly with inputs in different languages. In fact, using a non-English prompt for a non-English input does not provide superior results. However, when using two examples, each in a different language, the LLM mixes the languages in the JSON output. Hence, even when extracting data in multiple languages, users should consistently provide prompts and examples in one language.
- **Formats - Dates:** While invoices may display heterogeneity in date formats, the LLM output must contain only one standardised date format for consistency. While a generic instruction to use this format works for most documents (e.g., "Present dates in the following format: DD.MM.YYYY"), exceptions occur when converting from the MM/DD/YYYY format, which can be misinterpreted by the LLM as DD/MM/YYYY. Therefore, it is necessary to provide explicit instructions for this scenario (see Figure 5, line 5). Including this conversion issue in the example invoice also led to improved output. Trial and error confirmed that the use of examples like "If the date is 07/04/2022 enter 04.07.2022" is superior to using general instructions such as "convert from e.g., MM/DD/YYYY". It is also necessary to specifically remind the LLM at the end of the instruction section of the prompt about the date format conversion (see Figure 5, line 44). As our semantic data was created in Outlook format, there was no heterogeneity in dates. Hence, no examples were required.
- **Formats - Implicit Dates:** Payment conditions are not always given in the form of a date but can state "due on receipt" or "due in XX days", occasionally leading to wrong entity values. As a result, the LLM sometimes calculates the due date by adding the number of days stated on the invoice to the invoice date (Boye & Moell, 2025). However, this does not align with the actual due date. Instructions like "Never calculate!" were unsuccessful, regardless of their exact wording. Our iterative optimisation showed that a general instruction clarifying what to do in any case of uncertainty resolved the problem (see Figure 5, line 10).
- **Formats - Numbers:** To ensure consistent usage of number formats (e.g., "xx.xx" instead of "xx,xx") for entities such as the tax rate of the total price, a combination of the general instruction to present all numbers in the continental European number format "xx,xx" (see Figure 5, line 6) and entity-specific instructions in parentheses for those entities that hold numeric values (see Figure 5, line 30; Figure 5, line 31) proved to be the most effective. Only in combination with the trigger-word "strictly" (see Figure 5, line 31; see Guideline 3), the LLM consistently performs the transformation to the correct format.
- **Addresses:** Due to the use of various address formats in both the semantic and non-semantic datasets, extraction errors occur frequently (e.g., omitted house numbers or postal codes, incorrect numbers). This problem can be solved by

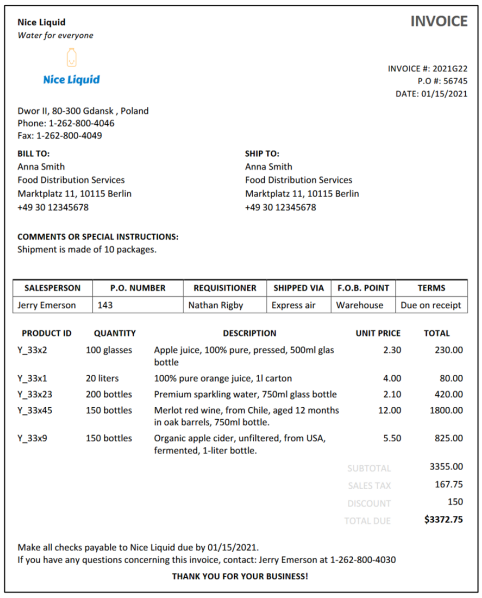
requiring the LLM to extract numeric values (i.e., house numbers) together with the relevant context (i.e., street names). For a correct SupPostalCode output, it is necessary to explain to the LLM that different postal code formats exist by naming examples (see Figure 5, line 22; Figure 6, line 16). Nevertheless, sometimes house numbers are missing if they are written in front of the street name, as in the USA's address format. In such situations, the output quality highly depends on the house number format and size. A house number value 4507 is more likely to be omitted than a value 86b. To prevent this phenomenon and ensure the address output is accurate, including an example addressing this issue (see Figure 7) is indispensable.

- **Multiple outputs:** For semantic data, the LLM frequently generates a second JSON output or hallucinates a completely new email after finishing the JSON. This phenomenon was only occasionally observed for non-semantic invoice data. Explicitly requiring the LLM at the end of the prompt to stop generating additional text once the task is completed, resolves this issue (i.e., "Stop the output after the JSON is completed and all entities are included. Do not provide any further explanation or information. Do not generate a new email.", see Figure 6, line 41). This solution is reinforced by providing examples that make it clear that only one JSON output is required. Another possible solution for addressing such hallucinations is to use a stopping sequence. For instance, starting and beginning the JSON with two curly brackets so that the stopping sequence can be set to "}}". No comparable action was required for non-semantic invoice data, as the lack of semantic context seems to remove triggers for the LLM to continue producing output.

Trouble Shooting

- **Interaction and backward testing:** In addition to interacting with the LLM to identify and improve upon prompt issues (White *et al.,* 2023), it is possible that during the iterative process, a prior prompt version delivered a better result than the current version. In this case, backward testing proved helpful. This involves deleting specific sections of the prompt until the problematic part is identified.

Input: Invoice (PDF)



Nice Liquid
Water for everyone

Nice Liquid

INVOICE

INVOICE #: 2021G22
P.O #: 56745
DATE: 01/15/2021

Dwor II, 80-300 Gdansk , Poland
Phone: 1-262-800-4046
Fax: 1-262-800-4049

BILL TO:
Anna Smith
Food Distribution Services
Marktplatz 11, 10115 Berlin
+49 30 12345678

SHIP TO:
Anna Smith
Food Distribution Services
Marktplatz 11, 10115 Berlin
+49 30 12345678

COMMENTS OR SPECIAL INSTRUCTIONS:
Shipment is made of 10 packages.

SALESPERSON	P.O. NUMBER	REQUISITIONER	SHIPPED VIA	F.O.B. POINT	TERMS
Jerry Emerson	143	Nathan Rigby	Express air	Warehouse	Due on receipt

PRODUCT ID	QUANTITY	DESCRIPTION	UNIT PRICE	TOTAL
Y_33x2	100 glasses	Apple juice, 100% pure, pressed, 500ml glas bottle	2.30	230.00
Y_33x1	20 liters	100% pure orange juice, 1l carton	4.00	80.00
Y_33x23	200 bottles	Premium sparkling water, 750ml glass bottle	2.10	420.00
Y_33x45	150 bottles	Merlot red wine, from Chile, aged 12 months in oak barrels, 750ml bottle.	12.00	1800.00
Y_33x9	150 bottles	Organic apple cider, unfiltered, from USA, fermented, 1-liter bottle.	5.50	825.00
		SUBTOTAL		3355.00
		SALES TAX		167.75
		DISCOUNT		150
		TOTAL DUE		**$3372.75**

Make all checks payable to Nice Liquid due by 01/15/2021.
If you have any questions concerning this invoice, contact: Jerry Emerson at 1-262-800-4030

THANK YOU FOR YOUR BUSINESS!

Input: Invoice (PDF) converted to TXT

Nice Liquid Water for everyone Dwor II, 80-300 Gdansk , Poland Phone: 1-262-800-4046 Fax: 1-262-800-4049 INVOICE INVOICE #: 2021G22 P.O #: 56745 DATE: 01/15/2021 BILL TO: Anna Smith Food Distribution Services Marktplatz 11, 10115 Berlin +49 30 12345678 SHIP TO: Anna Smith Food Distribution Services Marktplatz 11, 10115 Berlin +49 30 12345678 COMMENTS OR SPECIAL INSTRUCTIONS: Shipment is made of 10 packages. SALESPERSON P.O. NUMBER REQUISITIONER SHIPPED VIA F.O.B. POINT TERMS Jerry Emerson 143 Nathan Rigby Express air Warehouse Due on receipt PRODUCT ID QUANTITY DESCRIPTION UNIT PRICE TOTAL Y_33x2 100 glasses Apple juice, 100% pure, pressed, 500ml glas bottle 2.30 230.00 Y_33x1 20 liters 100% pure orange juice, 1l carton 4.00 80.00 Y_33x23 200 bottles Premium sparkling water, 750ml glass bottle 2.10 420.00 Y_33x45 150 bottles Merlot red wine, from Chile, aged 12 months in oak barrels, 750ml bottle. 12.00 1800.00 Y_33x9 150 bottles Organic apple cider, unfiltered, from USA, fermented, 1-liter bottle. 5.50 825.00 SUBTOTAL 3355.00 SALES TAX 167.75 DISCOUNT 150 TOTAL DUE $3372.75 Make all checks payable to Nice Liquid due by 01/15/2021. If you have any questions concerning this invoice, contact: Jerry Emerson at 1-262-800-4030 THANK YOU FOR YOUR BUSINESS!

Desired Output: Ground Truth (JSON)


```
{
      "InvoiceNo": "2021G22",
      "InvoiceDate": "15.01.2021",
      "ExpectedDeliveryDate": null,
      "SupCompany": "Nice Liquid",
      "SupContactPerson": "Jerry Emerson",
      "SupEmail": null,
      "SupPhone": "1-262-800-4046",
      "SupAddress": "Polanki 124D/3",
      "SupCity": "Gdańsk",
      "SupPostalCode": "80-308",
      "SupCountry": "Poland",
      "ProList": [
       {
        "ArticleNo": "Y_33x2",
        "Name": "Apple Juice",
        "Quantity": 100,
        "QuantityUnit": "glasses"
       },
       [...]
      ],
      "TaxRate": null,
      "TotalPrice": "3372,75",
      "CurrencySign": "$",
      "DueDate": "01.15.2021"
}
```


Figure 7. Sample data input (PDF and TXT format) and output

4.3 Limitations of LLMs for data extraction: unsolved challenges

However, some challenges with respect to the extraction of *semi-structured, non-semantic data* could not be overcome by prompt engineering. These issues were driven by the fact that semantic connections between tokens or phrases, for example in tables, are lost upon conversion of PDF to TXT documents (see Appendix), which strongly influences the quality of the extraction output (Nasar *et al.* 2021). Specifically, we faced the following conversion-related issues in our invoice dataset:

1) When converting a table containing both the *due date and the shipping date in close proximity*, the resulting TXT-file displayed the following: "Services Due Date: Shipping Date: 01/12/2021 01/05/2021". Thus, the LLM identified "01/12/2021" as the shipping date, although this in fact reflects the due date. Although only four documents in the invoice dataset were affected, this issue could not be solved by adjusting the prompt. As it is not even possible for human readers to identify the correct date based on the TXT-file, only better quality of the PDF-to-TXT-conversion could solve this issue.
2) *Numeric cell values* in non-semantic data sources may not be correctly identified if they do not have any additional cues with respect to their content (e.g., "€" or "EUR" to indicate a price; "kg" or "g" or "pcs" to indicate units) (see Figure 8).
3) When extracting *longer numeric values such as phone numbers*, the formatting may not be consistently extracted (e.g., 100 433-4534 instead of (100) 433-4534) or it extracts the phone and fax number as one piece of information if they are written directly one after the other without an indicator, such as (100) 433-4534| (100) 433-4542. Because phone number formats differ depending on the invoice layout, a pattern-matching approach is not feasible (Moundas *et al.,* 2024).
4) *Invoice numbers* were sometimes not identified if they were in purely numerical format and not attached to a keyword such as "ID". This issue did not occur if invoice numbers were in alphanumeric format.
5) When companies displayed a *slogan* close to their company name on an invoice, the LLM did sometimes not only extract the company name but additionally the slogan of the company (e.g., for company "GreenS" it extracted "GreenS Health is green life" as the company name). Again, as the PDF to TXT conversion does not preserve the structure of the original file and supplier names are unknown to the LLM, this issue could not be solved. However, adding a post-LLM-processing step comparing the LLM output to the company's master data, may be a solution.

In addition, the following non-conversion-related issues could not be consistently solved:

6) If the input provides *conflicting information* (e.g., "1 bag (holding 25 kg)" and "25 kg"), the LLM extracts the correct information but may not attach

it to the correct entity (e.g., LLM output: "Quantity: 1", "QuantityUnit: 25 kg bag" vs. ground truth: "Quantity: 25", "QuantityUnit: kg"). Such cases were not counted as errors in the evaluation because the information is still correct in terms of content. In an industry with a limited range of feasible units, stating allowed units in the prompt is expected to improve the output. However, given the large number of possible units in the food industry, this was not an option for the sample data.

7) In some instances, the LLM hallucinated by generating a *non-existent tax rate* (see Figure 5, line 30). Whenever the tax rate was not explicitly stated on the invoice, but an absolute value for the sales tax was given, the LLM either presented the absolute monetary value as a percentage or calculated an incorrect tax rate.

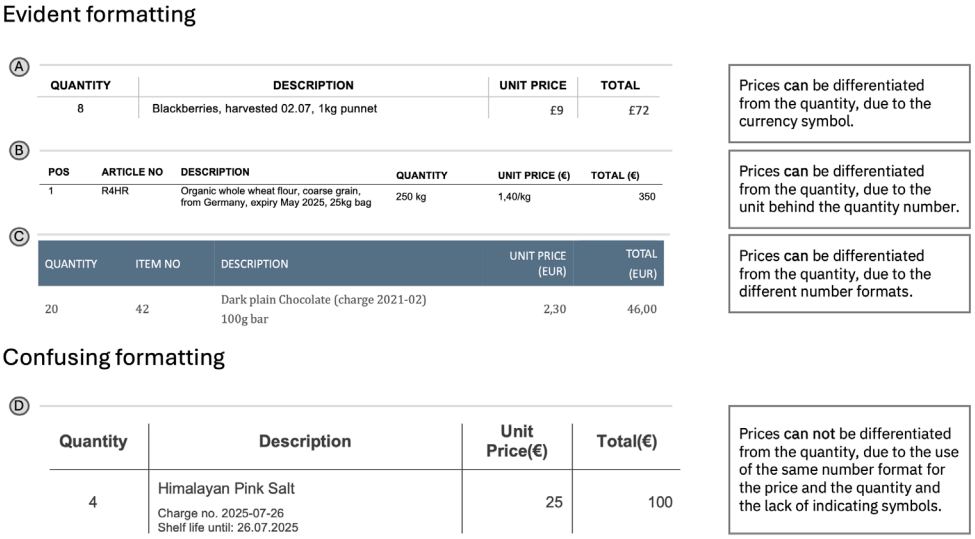


Figure 8. Relevance of formatting in invoice data extraction: Examples of formats and formatting issues

While all above issues related to our sample of semi-structured non-semantic data, order emails only displayed one major problem:
Unstructured, free-form semantic emails allow for the provision of non-standard information that may be relevant to the order (e.g., notification requests, requests for early delivery, or information about changes in the delivery address compared to the last order). When extracting this information into an entity called "AdditionalInfo", the LLM repeatedly paraphrased or omitted the information. Although paraphrasing may not necessarily be an issue, it makes it difficult to validate its correctness automatically in daily operations. While this is the only major issue for semantic data, this entity may contain information that is essential for correct order fulfilment.

These unresolved issues show how differences in the underlying data are ultimately a major driver of the differences in error rates of 2.5% (semantic) and 7% (non-semantic).

5. Discussion

In prompting, there are no standardised solutions, and each use case requires a unique prompt. Prompting is an iterative, time-consuming process and users must balance their time investment with output quality requirements. We show that prompts for data extraction from non-semantic invoice data and semantic email data differ substantially and lead to different quality outputs (error rates of 2.5% and 7% respectively). Furthermore, the prompt engineering task was substantially more time-consuming for non-semantic data because of the larger amount and difficulty of extraction challenges.

Our enhanced prompting guidelines and final prompts offer orientation for users within the accounting domain. For the specific case of data extraction from *invoices*, the final prompt is expected to yield results that can be entered into accounting systems. Output quality is in line with or even exceeds error rates achieved using other methodologies in prior literature (e.g., Ha & Horák, 2022). However, it is essential to check whether trigger words and example descriptions are valid in an out-of-sample context, to ensure that company-specific information and requirements are processed accurately.

Identified issues related primarily to the extraction of correct formats, such as phone numbers or the specific case of the tax rate. Some of these issues (e.g., phone numbers) do not constitute necessary inputs into the accounting process beyond the initial master data entries and may therefore be neglected. Despite these issues, using LLMs to extract data may still be an efficient and low-cost means of data extraction, even if post-extraction steps for automatically validating data based on other existing data sources (e.g., master data) or manual validation of critical cases may be necessary.

We expect that, with minor modifications, our prompt would work similarly well for other tabular data – potentially even better depending on the specific table structures and similarity of information that is placed in proximity. However, what matters most for output quality is the quality of table conversion from PDF to TXT and how much contextual information is lost in this process.

With respect to semantic but unstructured order emails, our prompt is expected to be transferable to a range of other semantic data sources. Data source-specific adaptations may be necessary to address the potential hallucination issues. If

systematic differences in order emails occur in a different setting, the prompt may be enhanced by adjusting examples (e.g., use the word "number" instead of "ID").

It is possible that some of the remaining problems could be better addressed by using a different LLM, as LLM outputs depend on the language and templates for which the models have been trained. However, there are limitations to the manual method of prompt engineering, even with few-shot prompting. Many studies (e.g., Ekin, 2023; Dang *et al.,* 2022; Knoth *et al.,* 2024; Li *et al.,* 2025; Zamfirescu-Pereira *et al.,* 2023) highlight the guesswork and experimentation characteristics of prompting and its limitations towards solving problems that use different input prompts, just like in the case of our analysis where the context given to the LLM differs with each invoice or email. A potential solution to this problem is prompt tuning. The LLM is tuned with a set of labelled data and task-specific samples called "soft prompt", which augments the actual prompt every time the model is asked to generate the output, thereby improving the output.

Another possible approach to address the unsolved problems is fine-tuning, which adapts the LLM to new tasks. In fine-tuning, the weights and parameters of a model's artificial neural network are changed using a labelled subject-specific dataset (e.g., invoices). This approach was used, for example, in the study by Hamdi *et al.* (2021). However, it requires considerably more time for manual labelling and more computing power, which incurs costs (Paaß & Giesselbach, 2023).

It is important to mention that GenAI may not be the best selection for some types of data. For instance, structured data, such as tables, can be handled better by traditional ML methods (Zewe, 2023). This is why Salgado and Sánchez (2023) suggest a hybrid approach combining multiple methods to achieve more robust results, as ML suffices for entity recognition and NLP techniques help to handle the semantic parts of documents.

6. Conclusion

LLMs may provide low-cost and efficient means without large investments in IT-infrastructure to automate the extraction of transactional accounting data inputs from heterogeneous sources. The low-cost nature, both in terms of implementation and running costs compared to established commercial software solutions, makes it an attractive option for SMEs that have limited resources available but are simultaneously under a lot of pressure to work more efficiently. Data sources used for data extraction may take the form of unstructured but semantic data (e.g., order emails) or (semi-)structured but non-semantic data that consists primarily of tabular data (e.g., invoices). The existing literature focuses on large-scale extraction of passages or individual data points from annual or sustainability reports. However, requirements with respect to extraction quality differ substantially from the

extraction of transactional data that will be fed into accounting systems and processes and, thus, have major implications for conducting business as well as for the audit process.

The output quality of LLMs is highly dependent on context, both with respect to the underlying data and the user instructions (i.e., prompt). Our proof-of-concept shows that general guidelines derived from prior literature for the extraction of large-scale data are not fully transferable to both our use cases. In addition, our findings show that the type of underlying data (i.e., semantic or non-semantic) determines both ease of extraction and output quality.

After identifying domain-specific challenges, we proposed solutions and adjusted the original literature-based guidelines accordingly. Specifically, we refine existing prompting guidelines with respect to "Communicate groundwork for understanding", "Structure the prompt", and "Define data extraction specifications". Most of these adjustments relate to the best location within the prompt, as well as specifying the use of examples and explanations. Furthermore, we identified some guidelines that are so specific to transactional accounting data extraction that they have not been addressed in the literature so far, namely "Multilingualism", "Formats – Dates/Numbers/Addresses", as well as hallucination of "multiple outputs".

Although users may directly apply our derived and optimised prompts with only very minor customisation, our guidelines are also applicable to other tabular or text-based accounting data extraction tasks. To keep the implementation effort and cost for businesses low, we tested our prompts using a standard pre-trained LLM without fine-tuning. Whether models created for financial documents such as annual reports (e.g., FinBERT (Huang *et al.,* 2023)) provide superior results in our context will require further research.

References

Akcali, Z., Cubuk, H. S., Oguz, A., Kocak, M., Farzaliyeva, A., Guven, F., Ramazanoglu, M. N., Hasdemir, E., Altundag, O., & Agildere, A. M. (2025) "Automated extraction of key entities from non-english mammography reports using named entity recognition with prompt engineering", *Bioengineering*, vol. 12, no. 2: 168, doi: 10.3390/bioengineering12020168

Bakarich, K. M., & O'Brien, P. E. (2021) "The robots are coming … but aren't here yet: The use of artificial intelligence technologies in the public accounting profession", *Journal of Emerging Technologies in Accounting*, vol. 18, no. 1: 27-43, doi: 10.2308/JETA-19-11-20-47

Beduschi, A. (2024) "Synthetic data protection: Towards a paradigm change in data regulation?", *Big Data & Society*, vol. 11, no. 1: 20539517241231277, doi: 10.1177/20539517241231277

Bergmann, D. (2024) "What is a context window?", IBM, available on-line at: https://www.ibm.com/think/topics/context-window

Bochkay, K., Brown, S. V., Leone, A. J., & Tucker, J. W. (2023) "Textual analysis in accounting: What's next?", *Contemporary Accounting Research*, vol. 40, no. 2: 765-805, doi: 10.1111/1911-3846.12825

Bose, P., Srinivasan, S., Sleeman, W. C., Palta, J., Kapoor, R., & Ghosh, P. (2021) "A survey on recent named entity recognition and relationship extraction techniques on clinical texts", *Applied Sciences*, vol. 11, no. 18: 8319, doi: 10.3390/app11188319

Boye, J., & Moell, B. (2025) "Large language models and mathematical reasoning failures", *arXiv preprint arXiv:2502.11574,* doi: 10.48550/ARXIV. 2502.11574

Cheng, X., Dunn, R., Holt, T., Inger, K., Jenkins, J. G., Jones, J., Long, J. H., Loraas, T., Mathis, M., Stanley, J., & Wood, D. A. (2024) "Artificial intelligence's capabilities, limitations, and impact on accounting education: Investigating ChatGPT's performance on educational accounting cases", *Issues in Accounting Education*, vol. 39, no. 2: 23-47, doi: 10.2308/ISSUES-2023-032

Cooper, L. A., Holderness, D. K., Sorensen, T. L., & Wood, D. A. (2019) "Robotic process automation in public accounting", *Accounting Horizons*, vol. 33, no. 4: 15-35, doi: 10.2308/acch-52466

Cui, L., Xu, Y., Lv, T., & Wei, F. (2021) "Document AI: Benchmarks, models and applications", *arXiv preprint*, doi: 10.48550/ARXIV.2111.08609

Dang, H., Mecke, L., Lehmann, F., Goller, S., & Buschek, D. (2022) "How to prompt? Opportunities and challenges of zero- and few-shot learning for human-AI interaction in creative applications of generative models", *arXiv*, doi: 10.48550/ARXIV.2209.01390

Ekin, S. (2023) "Prompt engineering for ChatGPT: A quick guide to techniques, tips, and best practices", *Authorea Preprints*, doi: 10.36227/techrxiv.22683919.v1

Emett, S., Eulerich, M., Lipinski, E., Prien, N., & Wood, D. A. (2025) "Leveraging ChatGPT for enhancing the internal audit process—A real-world example from Uniper, a large multinational company", *Accounting Horizons*, vol. 39, no. 2: 125-135, doi: 10.2308/HORIZONS-2023-111

Eulerich, M., & Wood, D. A. (2023) "A demonstration of how ChatGPT can be used in the internal auditing process", *SSRN Electronic Journal*, doi: 10.2139/ssrn.4519583

Eulerich, M., Sanatizadeh, A., Vakilzadeh, H., & Wood, D. A. (2024) "Is it all hype? ChatGPT's performance and disruptive potential in the accounting and auditing industries", *Review of Accounting Studies*, vol. 29, no. 3: 2318-2349, doi: 10.1007/s11142-024-09833-9

Föhr, T. L., Schreyer, M., Juppe, T. A., & Marten, K. U. (2023) "Assuring sustainable futures: Auditing sustainability reports using AI foundation models", *SSRN Electronic Journal,* doi: 10.2139/ssrn.4502549

Gemini Team Google (2024) "Gemini 1.5 unlocking multimodal understanding across millions of tokens of context", *Google DeepMind*, *arXiv preprint arXiv:2403.05530*, doi: 10.48550/arXiv.2403.05530

Gregor, S., & Hevner, A. R. (2013) "Positioning and presenting design science research for maximum impact", *MIS Quarterly*, vol. 37, no. 2: 337-355, doi: 10.25300/MISQ/2013/37.2.01

Ha, H. T., & Horák, A. (2022) "Information extraction from scanned invoice images using text analysis and layout features", *Signal Processing: Image Communication*, vol. 102: 116601, doi: 10.1016/j.image.2021.116601

Hamdi, A., Carel, E., Joseph, A., Coustaty, M., & Doucet, A. (2021) "Information extraction from invoices", In J. Lladós, D. Lopresti, & S. Uchida (Eds.), *Document Analysis and Recognition – ICDAR 2021* (vol. 12822: 699-714). Springer International Publishing, doi: 10.1007/978-3-030-86331-9_45

Hu, Y., Chen, Q., Du, J., Peng, X., Keloth, V. K., Zuo, X., Zhou, Y., Li, Z., Jiang, X., Lu, Z., Roberts, K., & Xu, H. (2024) "Improving large language models for clinical named entity recognition via prompt engineering", *Journal of the American Medical Informatics Association*, vol. 31, no. 9:1812-1820, doi: 10.1093/jamia/ocad259

Huang, A. H., Wang, H., & Yang, Y. (2023). "FinBERT: A large language model for extracting information from financial text", *Contemporary Accounting Research*, vol. 40, no. 2: 806-841, doi: 10.1111/1911-3846.12832

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025) "A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions", *ACM Transactions on Information Systems*, vol. 43, no. 2: 1-55 https://doi.org/10.1145/3703155

Ida, M. (2024) *A narrative history of artificial intelligence: The perpetual frontier of information technology*. Springer Nature Singapore, doi: 10.1007/978-981-97-0771-3

Kirstain, Y., Lewis, P., Riedel, S., & Levy, O. (2021) "A few more examples may be worth billions of parameters", *arXiv preprint arXiv:2110.04374*, doi: 10.48550/ARXIV.2110.04374

Klein, B., Dengel, A. R., & Fordan, A. (2004) "*SmartFIX: An adaptive system for document analysis and understanding*", In A. Dengel, M. Junker, & A. Weisbecker (Eds.), *Reading and Learning*. Berlin, Heidelberg: Springer.

Knoth, N., Tolzin, A., Janson, A., & Leimeister, J. M. (2024) "AI literacy and its implications for prompt engineering strategies", *Computers and Education: Artificial Intelligence*, vol. 6: 100225, doi: 10.1016/j.caeai.2024.100225

Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2022) "Large Language Models are Zero-Shot Reasoner", In *S.* Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.), *Advances in Neural Information Processing Systems 35*

Korzyński, P., Mazurek, G., Krzypkowska, P., & Kurasiński, A. (2023) "Artificial intelligence prompt engineering as a new digital competence: Analysis of generative AI technologies such as ChatGPT", *Entrepreneurial Business and Economics Review,* vol. 11, no. 3: 25-37

Lacity, M., & Willcocks, L. (2016) "A new approach to automating services", *MIT Sloan Management* Review. vol. 58, no. 1: 41-49

Li, H., & Vasarhelyi, M. A. (2024) "Applying large language models in accounting: A comparative analysis of different methodologies and off-the-shelf examples", *Journal of Emerging Technologies in Accounting*, vol. 21, no. 2: 133-152, doi: 10.2308/JETA-2023-065

Li, H., Gao, H., Wu, C., & Vasarhelyi, M. A. (2025) "Extracting financial data from unstructured sources: Leveraging large language models", *Journal of Information Systems*, vol. 39, no. 1: 135-156, doi: 10.2308/ISYS-2023-047

Li, J., Sun, A., Han, J., & Li, C. (2022) "A survey on deep learning for named entity recognition", *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 1: 50-70, doi: 10.1109/TKDE.2020.2981314

Lipenkova, J. (2022) "Choosing the right language model for your NLP use case", *Towards Data Science*, available on-line at: https://towardsdatascience.com/choosing-the-right-language-model-for-your-nlp-use-case-1288ef3c4929/?source=rss----7f60cf5620c9---4

Liu, F. (2025) "Deep feature extraction method for automatic classification and processing of accounting information", *IEEE Access*, vol: 13: 193232-193248, doi: 10.1109/ACCESS.2025.3625441

Liu, J., Shen, D., Zhang, Y., Dolan, B., Carin, L., & Chen, W. (2021) "What makes good in-context examples for GPT-3?", *arXiv preprint arXiv:2101.06804*, doi: 10.48550/arXiv.2101.06804

Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., & Neubig, G. (2023) "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing", *ACM Computing Surveys*, vol. 55, no. 9: 1-35, doi: 10.1145/3560815

Liu, V., & Chilton, L. B. (2022) "Design guidelines for prompt engineering text-to-image generative models", In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1-23, doi: 10.1145/3491102.3501825

Lu, Y., Bartolo, M., Moore, A., Riedel, S., & Stenetorp, P. (2021) "Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity", *arXiv preprint arXiv:2104.08786*, doi: 10.48550/ARXIV.2104.08786

Mistral AI team (2023) "Mixtral of experts: A high quality sparse mixture-of-experts", *Mistral AI*, available on-line at: https://mistral.ai/news/mixtral-of-experts

Mistral AI (2025) "Prompting capabilities", available on-line at: https://docs.mistral.ai/guides/prompting_capabilities/

Moffitt, K. C., Rozario, A. M., & Vasarhelyi, M. A. (2018) "Robotic process automation for auditing", *Journal of Emerging Technologies in Accounting*, vol. 15, no. 1: 1-10, doi: 10.2308/jeta-10589

Moundas, M., White, J., & Schmidt, D. C. (2024) "Prompt patterns for structured data extraction from unstructured text", In *Proceedings of the 31st Pattern Languages of Programming (PLoP) Conference*

Nasar, Z., Jaffry, S. W., & Malik, M. K. (2022) "Named entity recognition and relation extraction: State-of-the-art", *ACM Computing Surveys*, vol. 54, no. 1: 1-39, doi: 10.1145/3445965

Ni, J., Bingler, J., Colesanti-Senni, C., Kraus, M., Gostlow, G., Schimanski, T., Stammbach, D., Vaghefi, S. A., Wang, Q., Webersinke, N., Wekhof, T., Yu, T., & Leippold, M. (2023) "Chatreport: Democratizing sustainability disclosure analysis through LLM-based tools", *arXiv preprint arXiv:2307.15770*, doi: 10.48550/arXiv.2307.15770

Paaß, G., & Giesselbach, S. (2023) *Foundation models for natural language processing: Pre-trained language models integrating media*. Cham: Springer International Publishing, doi: 10.1007/978-3-031-23190-2

Peterson, K. (2012) "Accounting complexity, misreporting, and the consequences of misreporting", *Review of Accounting Studies*, vol. 17, no. 1: 72-95, doi: 10.1007/s11142-011-9164-5

Petković, D. (2017) "JSON integration in relational database systems", *International Journal of Computer Applications*, vol. 168, no. 5: 14-19

Polat, F., Tiddi, I., & Groth, P. (2025) "Testing prompt engineering methods for knowledge extraction from text", *Semantic Web: – Interoperability, Usability, Applicability*, vol. 16, no. 2: SW-243719, doi: 10.3233/SW-243719

Ray, P. P. (2023) "ChatGPT: A comprehensive review on background, applications, key challenges, bias, ethics, limitations and future scope", *Internet of Things and Cyber-Physical Systems*, vol. 3: 121-154, doi: 10.1016/j.iotcps.2023.04.003

Ruiz, A., Ashoori, M., & Bergmann, D. (2024) "Meta releases new Llama 3.1 models, including highly anticipated 405B parameter variant", *IBM*, available on-line at: https://www.ibm.com/blog/meta-releases-llama-3-1-models-405b-parameter-vari-ant/?utm_medium=OSocial&utm_source=Linkedin&utm_content=WXAWW&utm_id=IBMwatsonxLILLama31watsonx20240723&social_post=sf195764382&sf195764382=1

Salgado, A., & Sánchez, J. (2023) "Information extraction from electricity invoices through named entity recognition with transformers", In *5th International Conference on Advances in Signal Processing and Artificial Intelligence*

Saout, T., Lardeux, F., & Saubion, F. (2024) "An overview of data extraction from invoices", *IEEE Access*, vol. 12: 19872-19886, doi: 10.1109/ACCESS.2024.3360528

Schmidt, D. C., Spencer-Smith, J., Fu, Q., & White, J. (2024) "Towards a catalog of prompt patterns to enhance the discipline of prompt engineering", *ACM SIGAda Ada Letters*, vol. 43, no. 2: 43-51, doi: 10.1145/3672359.3672364

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

Shin, T., Razeghi, Y., Logan, R. L., Wallace, E., & Singh, S. (2020) "Autoprompt: Eliciting knowledge from language models with automatically generated prompts", *arXiv*, doi: 10.48550/ARXIV.2010.15980

Stryker, C., & Scapicchio, M. (2024) "What is generative AI?", *IBM*, available online at: https://www.ibm.com/topics/generative-ai (last accessed 23 July 2025)

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., & Lim, E.-P. (2023) "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models", *arXiv preprint arXiv:2305.04091*, doi: 10.48550/ARXIV.2305.04091

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022) "Chain-of-thought prompting elicits reasoning in large language models", In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, & A. Oh (Eds.)*, Advances in Neural Information Processing Systems 35*

White, J., Fu, Q., Hays, S., Sandborn, M., Olea, C., Gilbert, H., Elnashar, A., Spencer-Smith, J., & Schmidt, D. C. (2023) "A prompt pattern catalog to enhance prompt engineering with ChatGPT", *arXiv*, doi: 10.48550/ARXIV.2302.11382

Willcocks, L., Lacity, M., & Craig, A. (2015) "The IT function and robotic process automation", *The Outsourcing Unit Working Research Paper Series*, vol. 15, no. 5: 1-39

Wu, T., Terry, M., & Cai, C. J. (2022) "AI chains: Transparent and controllable human-AI interaction by chaining large language model prompts", In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, 1-22, doi: 10.1145/3491102.3517582

Wutzler, J. (2024) "Outsmarting artificial intelligence in the classroom—Incorporating large language model-based chatbots into teaching", *Issues in Accounting Education*, vol. 39, no. 4: 183-206, doi: 10.2308/ISSUES-2023-064

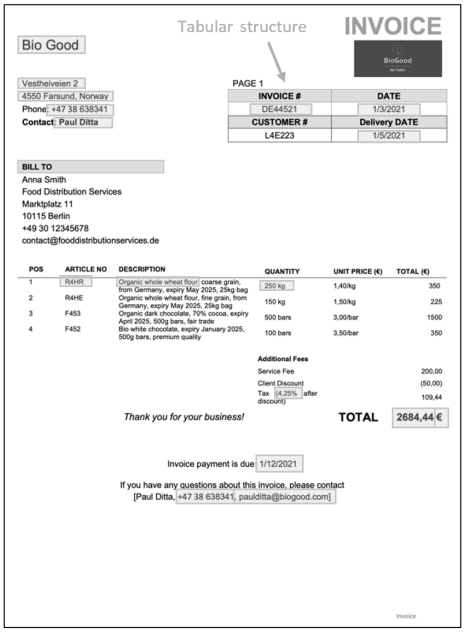
Zamfirescu-Pereira, J. D., Wong, R. Y., Hartmann, B., & Yang, Q. (2023) "Why Johnny can't prompt: How non-AI experts try (and fail) to design LLM prompts", *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, 1-21, doi: 10.1145/3544548.3581388

Zewe, A. (2023) "Explained: Generative AI", *Massachusetts Institute of Technology*, available on-line at: https://news.mit.edu/2023/explained-generative-ai-1109

Zhao, Z., Wallace, E., Feng, S., Klein, D., & Singh, S. (2021) "Calibrate before use: Improving few-shot performance of language models", In *Proceedings of the 38th International Conference on Machine Learning*

Appendix: BioGood sample invoice with lost tabular structure before and after PDF to TXT conversion

Invoice PDF



Converted to TXT

Tabular structure gets lost

Invoice Bio Good INVOICE Seestraße 4 PAGE 1 16503 Farsund INVOICE # DATE Phone: (755) 638-3416 DE44521 3/1/2021 Contact: Paul Ditta CUSTOMER # Delivery DATE L4E223 5/1/2021 BILL TO Anna Smith Food Distribution Services Marktplatz 11 10115 Berlin +49 30 12345678 contact@fooddistributionservices.de POS ARTICLE NO DESCRIPTION QUANTITY UNIT PRICE(€) TOTAL(€) 1 R4HR Organic whole wheat flour, coarse grain, from Germany, expiry May 2025, 25kg bag 250 kg 1,40/kg 350 2 R4HE Organic whole wheat flour, fine grain, from Germany, expiry May 2025, 25kg bag 150kg/2 bags 1,50/kg 225 3 F453 Organic dark chocolate, 70% cocoa, expiry April 2025, 500g bars, fair trade 500 bars/25kg 3,00/bar 1500 4 F452 Bio white chocolate, expiry January 2025, 500g bars, premium quality 100 bars 3,50/bar 350 Additional Fees Service Fee 200,00 Client Discount (50,00) Tax (4.25% after discount) 109,44 Thank you for your business! TOTAL 2684,44€ Invoice payment is due: 8/1/2021 If you have any questions about this invoice, please contact [Paul Ditta, 077124798, paulditta@biogood.com]