electronics

Article

Evaluation of Prompt Engineering on the Performance of a Large Language Model in Document Information Extraction

Lun-Chi Chen [1], Hsin-Tzu Weng [2], Mayuresh Sunil Pardeshi [3], Chien-Ming Chen [4], Ruey-Kai Sheu [5] and Kai-Chih Pai [1,*]

[1] College of Engineering, Tunghai Univeristy, Taichung 407224, Taiwan; lunchi@thu.edu.tw
[2] AI Center, Tunghai University, Taichung 407224, Taiwan; htweng@go.thu.edu.tw
[3] School of Computer Sciences, UPES, Dehradun 248007, India; pardeshi.mayuresh@ddn.upes.ac.in
[4] Department of Business Administration, National Taichung University of Science and Technology, Taichung 404336, Taiwan; jchen@nutc.edu.tw
[5] Department of Computer Science, Tunghai University, Taichung 407224, Taiwan; rickysheu@thu.edu.tw
* Correspondence: kcpai@thu.edu.tw

Abstract: The accelerated digitization of documentation, including paper invoices and receipts, has mitigated the necessity for precise and expeditious information management. Nevertheless, it has become unfeasible for humans to manually capture data due to the laborious and time-consuming nature of the process. The paper proposed a low training cost, prompt-based applied key information extraction (applied KIE) pipeline of the information extraction approach with Amazon Textract and Automatic Prompt Engineer (APE) using large language models (LLMs). A series of experiments were conducted to evaluate the performance of the proposed approach, with the results indicating an average precision of 95.5% and document information extraction accuracy of 91.5% on the SROIE (a widely used English dataset), and an average precision of 97.15% and a document information extraction accuracy of 85.29% on the invoice dataset from a Taiwanese shipping company.

Keywords: prompt engineering; large language models; key information extraction

Academic Editors: Marko Horvat and Tomislav Jagušt

Received: 19 April 2025
Revised: 15 May 2025
Accepted: 22 May 2025
Published: 24 May 2025

Citation: Chen, L.-C.; Weng, H.-T.; Pardeshi, M.S.; Chen, C.-M.; Sheu, R.-K.; Pai, K.-C. Evaluation of Prompt Engineering on the Performance of a Large Language Model in Document Information Extraction. *Electronics* **2025**, *14*, 2145. https://doi.org/10.3390/electronics14112145

1. Introduction

The integration of artificial intelligence (AI) with industry has led to substantial advancements in document processing. This development underscores the necessity for industrial machines to operate in conjunction with skilled human operators, thereby enhancing productivity and quality. In the context of maintenance management, the challenges associated with digitizing data are of paramount importance, particularly in the context of processing paper-based forms such as invoices and shipping documents. The advent of recent information technology (IT) applications has enabled companies to process large volumes of data daily, including scanned PDFs such as invoices and resumes. A significant proportion of these documents require manual processing to extract critical information, with as many as 70 files per day from over 30 different formats. Given a processing time of approximately five minutes per document, this inefficient workflow consumes about 5.83 h per day.

Extracting information from invoice documents has become increasingly complex with the advent of various automation technologies. These methods are designed to simplify traditional manual processes, reduce errors, and increase efficiency. Current technologies are centered around Optical Character Recognition (OCR) [1], natural language processing (NLP) [2], multimodal approaches, and advanced machine learning models [3]. However, each approach has its limitations, which can impact their overall effectiveness in real-world applications.

There are many ready-made OCR tools, including Amazon Textract, Google Cloud Vision, and Azure AI OCR. Many OCR tools can capture different layouts and key–value pairs. This is not suitable in practice. The real dataset has many formats and different values in each. The date format "Feb 22, 2024" needs to be converted to "2024/02/22" for Taiwan. If there is no fixed format, the output cannot meet the company's needs. There are studies on KIE tasks. The simplest way is to use regular expressions, but this is not flexible [4]. If there are many form configurations, the update method must be followed, which is inflexible.

As deep learning continues to evolve, an increasing number of NER-based methods are being developed that mark and classify each token. In recent times, KIE tasks have evolved to incorporate more than just text; additional information, such as images and layout coordinates, is being integrated as well. These models generally support a wide range of downstream tasks, but they require substantial amounts of labeled data for training. However, NER methods have limitations in terms of addressing noise issues and providing a unified expression for the identified tokens. They also demonstrate suboptimal performance in scenarios involving OCR errors or long entities [5]. LLM-based methods demonstrate superiority in handling OCR noise when compared to NER. KIE can be executed with Open QA, where the start and end of the answer are output. In contrast, NER classifies each token. Although tolerance for OCR noise exists, it is incapable of resolving the noise problem in practical datasets, and there is an absence of a unified expression method for found tokens. QA models lack the capacity to reason about unknown concepts. To illustrate, if the container packing method is $4 \times 20'$, the model would not know that $20'$ is FT and $4'$ is the quantity. Additionally, substantial training data are necessary for these models. Real company datasets are noisy, and previous studies did not consider inference requirements.

To address these issues, we propose an information retrieval method based on OCR tools integrated with LLMs. This method utilizes the pure text obtained by the OCR tool, thereby reducing the image processing time in the multimodal model. Notably, users are only required to employ natural language to input requirements. The potential exists for the utilization of large language models with reasoning and learning capabilities to construct an information extraction method based on an LLM, which could be employed in enterprise practice [6]. This approach not only addresses the challenge posed by variable field and item formats across files but also provides the final output in the desired format and facilitates unit conversion, thereby streamlining the process of matching company data to the corresponding system fields. In instances where a fixed format is not available, the system offers reference values to assist users in making informed decisions.

The motivation for the applied KIE is to determine the optimal method for extracting data from invoices and responding to user queries as part of an LLM. In 2021, K. Yindumathi, S. Chaudhari, and R. Aparna published a paper on structured data extraction using machine learning from invoices [7]. The authors describe a method for extracting structured data from unstructured invoices using OpenCV, logistic regression, and the K-neighbors algorithm for cropping OCR images. However, they conclude that the accuracy and time required for this process are insufficient for real-world applications. It is therefore evident that a new model is required in order to overcome the aforementioned challenges and to respond to user queries on the invoices in an efficient manner.

Other studies also illustrate the evolution of document understanding from specialized multimodal transformers to flexible, prompt-driven LLM pipelines: Tang et al. introduced a unified encoder–decoder Vision Text Layout Transformer pre-trained on supervised financial report layouts to achieve state-of-the-art QA, extraction, and classification accuracy [8]. Kim et al. proposed an OCR-free Document Understanding Transformer pre-trained on synthetically generated business cards, receipts, and handwriting, using GPT-3 downstream to deliver high F1-scores and accuracy [9]. Huang et al. built on LayoutLMv3's multimodal embeddings and word-patch alignment to support forms, receipts, and document QA with strong F1, mean average precision, and accuracy and seamless integration with any GPT-style model [10].

The previous research, such as LayoutLMv3 and DocLLM, has focused on fine-tuning specialized architecture for document understanding. They require a lot of labeled data and may not be adaptable to different document formats for industry use. The present study presents a KIE pipeline, where flexibility and low integration cost are highly valued. The pipeline does not require a lot of various training datasets and can quickly process noisy, multi-format industrial data with high accuracy.

The main contributions of this paper include the following four areas:

1. Automatic information extraction from the source data collection: Intelligent document processing requires all the data to be extracted from the source portable document format (pdf) by KIE. Initially, the pdf field's position is identified to help to extract the information by the OCR. These operations can be further improved if the pdf document identification can be known prior, e.g., lading, invoice, bills, etc. Primary data consist of document title, date, contents, notes, etc.

2. To overcome noise issues for the document quality and consistency: Preprocessing is necessary to handle the noise issues within the pdf, while extracting text, tables, and forms. Normalizing the data can improve the data extraction and storage process for further embedding. In contrast, the location of the fields can be performed by knowing image registration issues. Nevertheless, data overlapping with stamps and signatures is also important. Robustly processing the several types of data format is one of the key constraints and is overcome by Amazon Textract.

3. To provide performance evaluation and unify the output format content systematically: The data extracted from multiple fields are stored in a structured JSON format. Therefore, multiple pdfs of the same category can be stored in a single file, i.e., order, receipts, invoices, etc. These JSON structures can then be helpful for the LLM to extract query-based information for output results. The results are presented using precision and confidence distribution of GPT-based zero-shot, one-shot, and few-shot learnings with manual prompt, IPC, and applied KIE Prompt. The model configuration also includes intent-based prompt calibration (IPC), manual prompt, and human response comparison for detail analysis.

4. To ensure complete response by the field prompt: A large language model (LLM) is necessary to use for logical handling for the output as field extraction. The detail analysis performed by APE with few-shot learning on the JSON data structure can help to provide an accurate response from the processed data. Therefore, the applied KIE algorithm simplifies the process for pdf information extraction.

2. Related Works

We conducted a thorough review of recent advances in document AI tools in document processing. These tools use advanced technologies to read, understand, and process documents, thereby improving the efficiency and accuracy of unstructured data processing. While these tools have significant potential, several challenges remain unresolved, which motivates our proposed approach.

In document processing, a study presented a method for low-level structure segmentation based on visual prompting by Liu et al. [11]. Tunable parameters are employed for the extraction of visual content in each image, which encompasses the segmentation of regions, the identification of objects, and the detection of out-of-focus pixels. However, their method relies heavily on tunable parameters, which may require manual adjustments for each document type, limiting scalability [11]. A visual document assistant for high-resolution data was presented by Liu et al. [12]. The approach combines visual document understanding with multi-model LLMs, which consist of instructions and content filtering, thereby enhancing efficiency. Subsequently, the encoding is applied using the Swin Transformer. Appalaraju et al. presented a method for understanding documents using local features, an encoder–decoder transformer that is capable of processing language, vision, and spatial features. This approach improves pre-training efficiency but does not address the integration of diverse document layouts [13].

In addition, Li et al. [14] demonstrated self-supervised pre-training for layout analysis and text recognition. While effective for basic layout analysis, it lacks the ability to dynamically adapt to complex and heterogeneous table structures. The application of vision-based document processing for the performance of AI tasks, including layout analysis, the detection of table data, image classification, and OCR-based text detection on unlabeled documents affect the model's performance. The review of legal documents through text classification was demonstrated by Mahoney et al. [15]. The use of explainable AI (XAI) response snippets for document decision classification allows for the location of case-related documents, thereby enhancing the overall speed and quality of the document review process. The presentation of an ad hoc document retrieval system using a neural passage model was presented by Ai et al. [16]. Multiple passages are weighted based on granularities to help rank the responses. A convolution process is used with a fusion network to aggregate information. Huang et al. [17] presented meta visual prompting with diversity. An optimized meta-prompt with a divide-and-conquer approach improves performance and converges with other prompts faster. This work focused on adapting downstream tasks by pre-trained vision-based prompting for data diversity.

The efficacy of information extraction in document extraction systems is significantly influenced by the choice of prompt optimization technique. Recent studies have identified a number of strategies that enhance performance. For example, Sun et al. [18] demonstrated the automated implementation of prompt optimization with hints. The prompt optimization method may be applied to any prompt that adapts to the residual sampling summarization paradigm and is suitable for the GPT-4 model. The objective of the Auto-hint method is to generate hints for the augmented prompt through the use of mini-batch stochastic gradient descent (SGD). The enrichment of prompts and the selection of labeled data based on chain-of-thought (CoT) are demonstrated by Shum et al. [19]. The process entails the optimization of CoT for the augmentation, pruning, and subsequent selection stages of training and inference. The developed method employs a variance-reduced policy gradient for arithmetic, symbolic, common sense, and non-reasoning tasks. Automatic prompt generation using language models is presented by T. Shin et al. [20]. The automatic prompt, which is based on pre-trained masked language models, is utilized for text entailment and sentiment analysis in scarce data. The auto-prompt model is based on gradient search and can be applied to GPT-3.

In a recent study, Zhou et al. [21] demonstrated the ability of LLMs to perform at a human level in prompt engineering tasks. The Monte Carlo search method is employed for black-box-based optimization problems. Zero-shot learning still achieves human-level performance, which is optimized over the instruction candidate pool for maximizing the scoring function. However, their approach is sensitive to initialization, leading to variability in results across tasks. Gao, Fisch, and Chen [22] presented a method for developing better few-shot learners for pre-trained language models. The automatic prompt search with the fine-tuning method is used with selected input context tasks. This few-shot learning based model performs better on NLP tasks for regression and classification.

Multimodal document understanding by layout based generative models was demonstrated by Wang et al. [23]. A bounding box is used to capture the document layout structure with text and spatial data by transformer and thus provides LLM-based generative reasoning. Document information extraction by in-context learning was presented by He et al. [24]. Document information extraction (DIE) is performed by an LLM to understand positional relationships. Document image understanding by text and layout pre-training is demonstrated by Xu et al. [25]. Document images are scanned by a pre-trained framework for textual and layout information together, which uses multi-label document classification and a masked visual language model. Multi-model inputs are taken including layout, token, and image embedding for better performance. Gradient descent and beam search for prompt optimization were presented by Pryzant et al. [26]. Prompts are automatically improved by the textual gradients relating to LLM training data. This model overcomes the limitations of discrete optimization and does not require any tuning. Implementing a prompt engineer model was demonstrated by Ye et al. [27]. Automatic prompt optimization is performed by a meta-prompt containing context specification, detailed descriptions, and a reasoning template. Thus, a diverse language task model with a versatile nature is achieved while improving performance. Optimization using LLMs was presented by Yang et al. [28]. Optimization as prompting is utilized in LLMs by first evaluating using the traveling salesman problem (TSP) and heuristic algorithms. Even though the optimized prompt outperforms recent benchmarks, still the challenges of initialization sensitivity and utilizing training error cases for optimization are present. The generation of label sequence for sequence-to-sequence model prompt was demonstrated by Yu et al. [29]. Auto-prompting with sequencing helps to overcome fine-tuned, pre-trained models by label sequences, beam search, and contrastive re-ranking for increasing the prediction space. Refine AI generative art by prompt editing was presented by Wang et al. [30]. This work focuses on text prompt improvement to obtain the correct expression for the generated image. Emotion expressions as feature effects are recognized by the grammar with nouns, verbs, adjectives, etc. Cross-domain sequential recommendation by auto-prompting was demonstrated by Guo et al. [31]. The method functions by domain-invariant and domain representations, where learning is resorted and a pre-trained sequence encoder and dual-learning are recommended enhancements. Robust transfer learning by black-box visual prompting was presented by Oh et al. [32]. The black-box visual prompting adapts to fine-tuned pre-trained models by coordinator and stochastic approximation with corrected gradient for simultaneous perturbation as a Zoo algorithm. Prompting for image caption control is demonstrated by Wang et al. [33]. A group of prompts are designed for pre-trained image captioner fine-tuning. Also, the prompts are optimized for word embedding space continuously with learnable vectors.

LLMs using compositional CoT prompting was demonstrated by Mitra [34]. In this work, a zero-shot prompting method is applied that uses scene graphs for obtaining LLMs compositional knowledge. Therefore, a stack of different objects can be explained by scene graphs and natural language-based responses. Visual prompting understanding and improvement by label mapping was presented by Chen et al. [35]. The iterative label mapping and visual prompting is used to remap the source to target labels automatically for drastically improving task accuracy. Thus, a visual prompting problem is solved by bi-level optimization across 13 datasets. Image restoration prompting with perception was demonstrated by Wang et al. [36]. The prompt restorer performs image restoration and degradation before prompting. However, this modulator uses a pre-trained model with a self-attention mechanism and gated degradation perception propagation for the restoration process. News claim fact verification by an LLM-based prompting method is presented by Zhang and Gao [37]. LLMs with in-context learning are used with four-shot demonstration and a hierarchical prompting method improves performance for user QA. Also, the reasoning capabilities for explanation provide better understanding. Video continuous learning by prompting was demonstrated by Villa et al. [38]. A deep neural network with pre-trained models for learning from videos and avoids forgetting in video class incremental learning. Expressive prompt learning with a vision transformer was presented by Das et al. [39].

Prompt learning is modified for vision transformer (ViT) integration for learning tokens from ViT and residual learnable tokens. This model is applied in few-shot segmentation and image classification. Directional stimulus prompting applied in guided LLMs was demonstrated by Li et al. [40]. Keywords are used as hints for the LLM's desired outcome with the optimizations by supervised fine-tuning and reinforcement learning. The application includes QA, summarization, and CoT reasoning. Improvement in LLM reasoning by hint prompting was presented by Zheng et al. [41]. Previous interaction-based answers serve as hints for the correct responses. It has phenomenal task improvements on math reasoning and can be combined with CoT for higher performance. Soft prompting of language models and vision for text optimization was demonstrated by Bulat et al. [42]. Language-Aware Soft Prompting (LASP) is used to reduce overfitting, with a group of optimizations for subset prompts and recalibration to align visual language and improve robustness.

Emotion-aware embedding fusion for LLM prompting was demonstrated by Rasool et al. [43]. In this work, hierarchical fusion and attention mechanisms are applied to integrate semantic and emotional features from therapy session transcripts. Multiple emotion lexicons (e.g., NRC, VADER, SentiWordNet) are combined with LLMs such as Flan-T5, Llama 2, DeepSeek-R1, and ChatGPT-4. Through emotion-aware embedding construction and context retrieval using FAISS vector search, the LLMs are enabled to generate empathetic, coherent, and contextually relevant responses, particularly for psychotherapy chatbot applications.

Although many research studies have proposed ways to improve the extraction of receipt or invoice information, including NLP-based processing, visual recognition, LLM feature engineering, etc., there are fewer evaluations of optimization methods and correction of recognition error fields using real industrial data. Therefore, the main contribution of this research is to integrate Optical Character Recognition (OCR) with a Large Linguistic Model (LLM) for robust data acquisition. Our research overcomes the limitations in the literature survey as follows:

- Practical approach for the application of LLMs and KIE applied in the industry.
- Algorithm and detail flowchart process disclosed for implementation.
- Use the industrial dataset for the invoices, lading bills to be processed.
- Experiments and analysis based on multiple learning schemes.

3. Materials and Methods

3.1. Applied KIE Pipeline Overview

Figure 1 illustrates the applied KIE pipeline and its workflow for processing PDF documents. The proposed framework operates as follows:

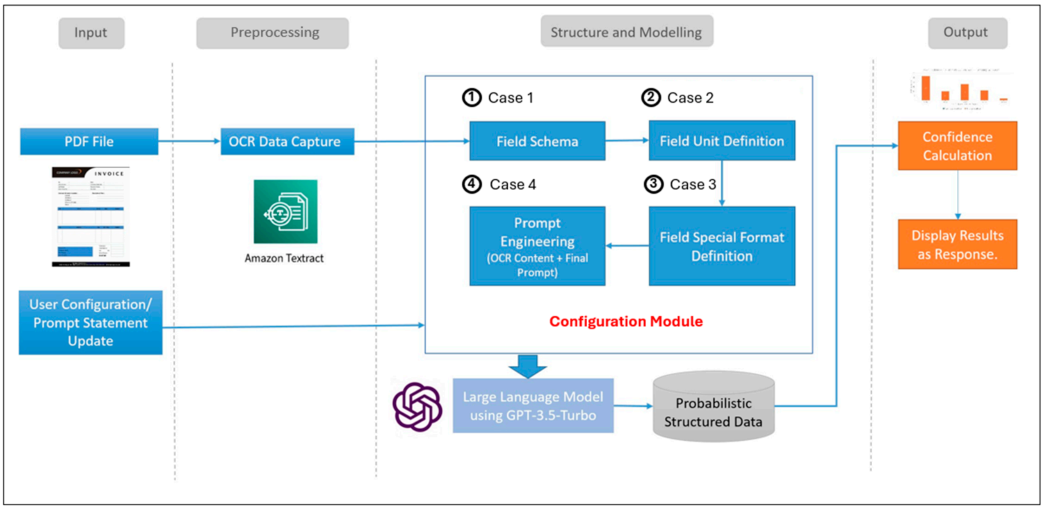

Input
Preprocessing
Structure and Modelling
Output
PDF File
OCR Data Capture
Amazon Textract
User Configuration/ Prompt Statement Update
① Case 1
② Case 2
Field Schema
Field Unit Definition
④ Case 4
③ Case 3
Prompt Engineering (OCR Content + Final Prompt)
Field Special Format Definition
Configuration Module
Large Language Model using GPT-3.5-Turbo
Probabilistic Structured Data
Confidence Calculation
Display Results as Response.

Figure 1. Applied KIE pipeline.

Input: The system receives a PDF document as an input. The language model (LLM) guidance mechanism takes a single input, the prompt instruction statement.

Document Preprocessing: The document is scanned, and its textual and structural data are extracted using Amazon Textract, an Optical Character Recognition (OCR) tool that ensures all content is captured accurately. The AnalyzeDocument API (Forms + Layout) was used to extract key–value pairs along with their layout coordinates, which helped preserve the structural integrity of the document during downstream LLM prompting.

As the structured information extracted by Amazon Textract does not align with the key information requirements of the company, the primary focus of this study is to output the content of the scanned file as unstructured text. This text will subsequently serve as the input for extracting key information in the subsequent stage of the prompting project using an LLM. Before the document information was extracted, we used the Python package anonymizedf (version 1.0.1) to protect sensitive data in a shipping invoice, especially the Identity and Contact Information (such as names, addresses, emails) and Payment and Banking Details (such as bank account numbers, bank names, letter of credit numbers).

Configuration Table Creation: For structure and modeling, a configuration table is created, including the following:

- Field Schema (Case 1): This field specifies field names and data types (e.g., string, integer).
- Field Unit Definition (Case 2): This field defines how field values are recorded (e.g., mg, kg, mt).
- Field Special Format (Case 3): This field specifies industrial details for field contents (e.g., "8 × 20 container" refers to an 8-foot width and 20-foot packaging height).

These configurations ensure that the extracted data are correctly structured for further processing.

Prompt-Based Data Structuring: The OCR-extracted content and the configuration table are processed through a user-defined prompt, thereby ensuring that the data are structured according to the configuration settings.

LLM Processing: The structured prompt is fed into an LLM, which refines the information into a final structured JSON output. The processed data are then stored in a data base for logging and query use.

Rule-based Post-processing for Container Count Fields: To enhance the extraction accuracy for fields requiring implicit reasoning, such as 20FT CONTAINERS NO and 40FT CONTAINERS NO, we implemented a lightweight rule-based post-processing module.

The post-processing module parses the LLM-generated textual output and applies regular expression matching to extract structured container counts.

Data Filtering and Confidence Calculation: For the results output, the system filters the structured data to remove noise values and irrelevant content. A confidence score is calculated using a predefined formula to measure the reliability of the results.

Query Handling: The field prompt enables user-specific queries to extract targeted information from the processed data. The system responds with structured output based on the query and the extracted data.

3.2. Mathematical Model

The user input transformation module that performs OCR-based extracted data transformation to the XML schema includes the user defined field name $F = \{F_1, F_2, \ldots, F_N\}$, with dtype $D = \{D_1, D_2, \ldots, D_N\}$. Next, the contents of target JSON schema $R = \{R_1, R_2, \ldots, R_N\}$, where N represents the number of captured fields, combined into JSON schema $JSON_S = \{F + D + R\}$. In such cases, the example $JSON_{EX} = \{E_1, E_2, \ldots, E_N\}$ corresponds to each field. For the field whose dtype is number, define its unit $U_M = \{U_1, U_2, \ldots, U_M\}$, where M represents the field of number output, which can be equal to 0. For the fields whose dtype is string, define their special requirements or format $S_K = \{S_1, S_2, \ldots, S_K\}$, where K represents the string output field, which can be equal to 0 and, $M + K$ does not necessarily need to equal N.

If the unit U_M or the special requirement S_K is not empty, then it can be changed to the format $T_{auto} = \{t_1, t_2, \ldots, t_{M+K}\}$ where $t_{m+k} = \forall \text{ field} : \{U_M \mid S_K\}$. Applying the input of automatic prompt engineering (APE), the LLM can generate the target user requirement prompt after understanding the content.

$$p(Y_{auto}|T_{auto}) = \textstyle\sum_1^{M+K} P_{lm}(y_{m+k}|t_{m+k}) \tag{1}$$

where T_{auto} is the preliminary definition of units and special needs, $m + k$ is the number of units and special needs, and the prompt Y_{auto} generated by the P_{lm} goal is obtained by maximizing the conditional probability.

Intuitive explanation for Equation (1): Equation (1) models the generation of auxiliary prompts describing units or special requirements for each field, based on maximizing the likelihood of correct field descriptions given T_{auto}.

During the meantime, the above information is combined to obtain the final requirement prompt.

$$C_{req} = \text{CONCAT}(JSON_S, JSON_{EX}, Y_{auto}) \tag{2}$$

Intuitive explanation for Equation (2): Equation (2) consolidates the structural definitions, examples, and auxiliary field prompts into a single, comprehensive user instruction-prompt.

Successively, for the information extraction module, we combine the final requirement prompt C_{req} from the user requirement transformation module with the file context T_{ocr} from OCR to obtain the content composition prompt.

$$C_{content} = \text{CONCAT}\big(C_{req}, T_{ocr}\big) \tag{3}$$

Intuitive explanation for Equation (3): Equation (3) merges the user requirement prompt with the document content extracted from OCR, forming the full input prompt used for downstream key information extraction.

$$p(Y_{KIE}|C_{content}) = \textstyle\sum_{l=1}^{L} P_{lm}(y_l|C_l) \tag{4}$$

Finally, $C_{content}$ is given as the content composition prompt, where l is the number of PDF files, and the LLM P_{lm} goal is obtained by maximizing the conditional probability to generate JSON Y_{KIE} that contains key information extraction. Additionally containing Y_{KIE} as a JSON output, in which each token generated has a probability value $P = \{p_1, p_2, \ldots, p_i\}$ with i is the total number of tokens generated in a single time. We add the probability of each document to obtain $P_{KIE} = \{P_1, P_2, \ldots, P_l\}$.

Intuitive explanation for Equation (4): Equation (4) models the LLM generating key information extraction (KIE) outputs by optimizing the probability of correct field retrievals based on the merged user prompt and OCR-extracted document content.

3.2.1. Applied Prompt Engineering (APE) for Prompt Optimization

LLMs provide a useful platform for the synthesis of a natural language program. Initially, population χ is considered for dataset $D_{train} = \{(Q, A)\}$ with tasks for input/output by prompt model M. When the input is given to the model M with query combination as [ρ:Q], then instruction ρ should be found with output A. Therefore, the optimization problem includes instruction ρ for $f(\rho$, Q, A), an expectation that maximizes the score on (Q, A) as follows:

$$\rho * = arg_p \max f(p) = arg_p max\ \mathbb{E}_{(Q,A)}[f(\rho, Q, A)] \tag{5}$$

Intuitive explanation for Equation (5): Equation (5) formalizes the automatic prompt engineering (APE) objective: selecting the prompt instruction ρ that yields the highest expected performance over the dataset, effectively searching for the most effective guidance for the model.

Here, the prompt ρ is optimized to produce output {A}. The automatic prompt engineer (APE) used for the automatic generation of instruction and selection [14] is an algorithm that chooses the best instruction for maximizing the score function. The APE algorithm is trained as $D_{train} \leftarrow \{(Q, A)\}_n$ and $f : \rho \times \mathrm{D} \longmapsto \mathbb{R}$ as a score function. Next, $\mathcal{U} \longleftarrow \{\rho_1, \ldots, \rho_m\}$ is used to sample multiple instructions by the LLM. A convergence function is utilized with a random $\tilde{D} \subset D_{train}$ data subset for evaluating its score $\tilde{s} \longleftarrow f\left(\rho,\ \tilde{D}_{train}\right)$. Similarly, for all other data subsets, the highest score instruction $\mathcal{U}_k \subset \mathcal{U}$ is obtained and updated with the LLM by resampling $\mathcal{U} \longleftarrow resample(\mathcal{U}\ _k)$. Finally, the $\rho * \leftarrow \mathrm{arg}max_{\rho \epsilon \mathcal{U}_k} f(\rho, D_{train}\ )$ is returned as the highest score instruction.

APE resample (U_k) is a data optimization technique that operates at the full prompt candidate level, whereas traditional beam search expands sequences token-by-token. It samples multiple complete prompts, scores them on a subset of data, and retains top candidates in a one-shot selection process, thus avoiding sequential decoding steps.

3.2.2. Instruction Prompt Calibration (IPC)

In addition to APE, we use an Instruction Prompt Calibration (IPC) mechanism to further refine prompt selection quality by integrating human and LLM-based evaluations [44]. Each candidate prompt ρ is evaluated based on two scores:

$S_{human}(\rho)$: A human-annotated correctness score.

$S_{LLM}(\rho)$: An LLM-evaluated consistency score.

The final IPC score for a prompt candidate is calculated as a weighted combination of these two scores:

$$S_{IPC}(\rho) = \alpha \times S_{human}(\rho) + (1 - \alpha) \times S_{LLM}(\rho) \tag{6}$$

where α ∈ [0, 1] is a weighting factor that balances human judgment and LLM evaluation (e.g., α = 0.7). The optimal prompt under IPC is selected as follows:

$$\rho^*_{IPC} = arg_{\rho} max S_{IPC}(\rho) \tag{7}$$

This IPC strategy allows the system to leverage both human expertise and LLM automation, ensuring that the prompts selected are robust not only for model bias but also for varying document contexts.

3.2.3. Confidence Calculation Module

In this confidence calculation module, we use the probability values $P_{KIE} = \{P_1, P_2, \ldots, P_l\}$ obtained after extracting the training dataset information. The field extraction confidence score is mapped to a range of 0–100 based on historical probability extremes. Specifically, the highest and lowest probability values observed from previous similar document sources (e.g., shipment invoices across different voyages) are recorded as dynamic upper and lower bounds.

New probability outputs are mapped linearly based on these historical extremes:

- If a new value falls within the existing bounds, it is scaled accordingly.
- If a new value exceeds the historical maximum or falls below the historical minimum, the bounds are updated and the confidence is mapped to 100 or 0, respectively.

This strategy ensures stable, interpretable confidence values without the need to manually remove outliers or rescale each batch independently.

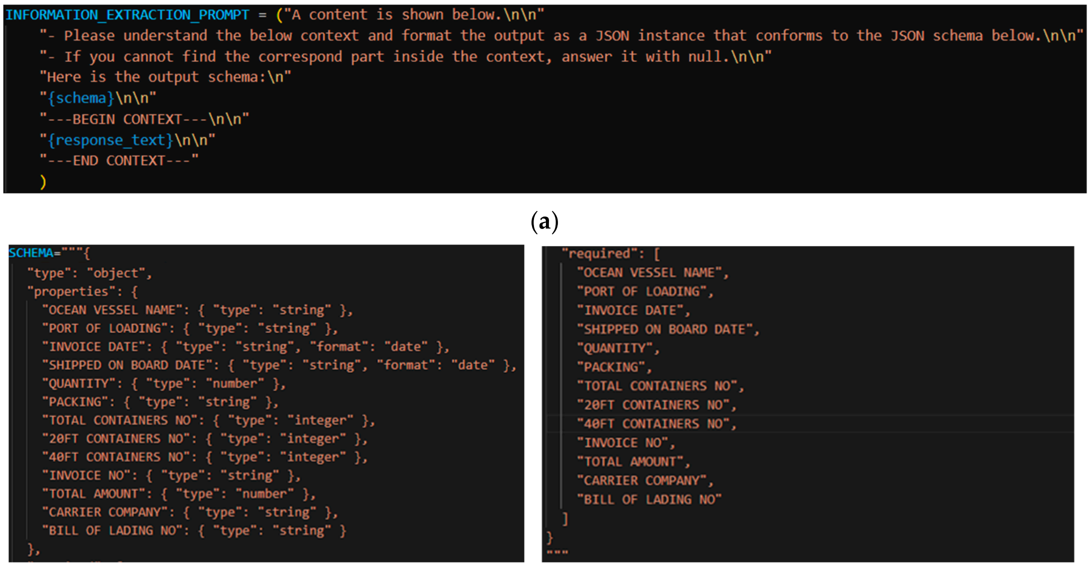
$$\begin{aligned} P'_{min} &= \lfloor P_{min} \rfloor \\ P'_{max} &= \lfloor P_{max} \rfloor \end{aligned} \tag{8}$$

Correspond these two numbers to the mapping interval of 0–100 to obtain the confidence calculation formula. The probability that is expected to be converted is P_{val} to obtain the final probability value S_i.

$$S_i = \frac{(P_{val} - P'_{min})}{P'_{max} - P'_{min}} * 100 \tag{9}$$

3.3. Prompt Contents and Design

In prompt modeling shown in Figure 2, the zero-shot prompt is used to extract field information with values from the input pdf. The prompt content specifies "What are the necessary processing required to be performed by this prompt". The first line in Figure 2a specifies the prompt to check for the contents. Next, the prompt is requested to understand the input and respond in the JSON schema format for the captured data. In this case, if the data are not available, then NULL values are responded. Later, the output schema specifies the beginning and end point. Figure 2b contains the field properties specifying the details of type and object. Here, all the required fields need to be specified in detail with data type and sequence for the fields to be given as output. In contrast, in Figure 3, the one-shot prompt learning consists of the new settings given for the prompt contents and Figure 3b shows the one-shot prompt with one example of the complete output that is taken from one pdf document.



```
INFORMATION_EXTRACTION_PROMPT = ("A content is shown below.\n\n"
    "- Please understand the below context and format the output as a JSON instance that conforms to the JSON schema below.\n\n"
    "- If you cannot find the correspond part inside the context, answer it with null.\n\n"
    "Here is the output schema:\n"
    "{schema}\n\n"
    "---BEGIN CONTEXT---\n\n"
    "{response_text}\n\n"
    "---END CONTEXT---"
    )
```

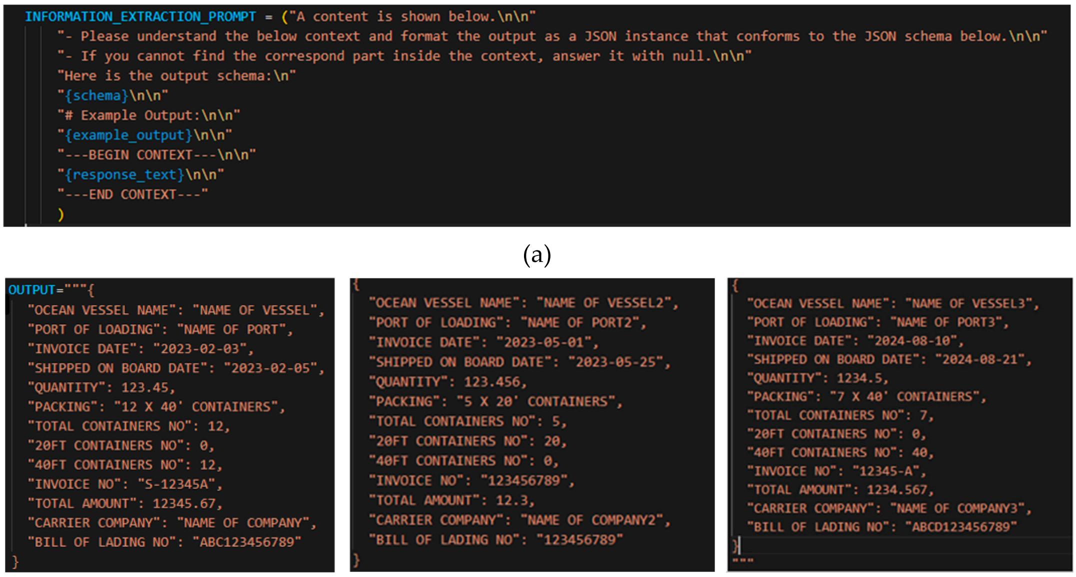
(**a**)

```
SCHEMA="""{
  "type": "object",
  "properties": {
    "OCEAN VESSEL NAME": { "type": "string" },
    "PORT OF LOADING": { "type": "string" },
    "INVOICE DATE": { "type": "string", "format": "date" },
    "SHIPPED ON BOARD DATE": { "type": "string", "format": "date" },
    "QUANTITY": { "type": "number" },
    "PACKING": { "type": "string" },
    "TOTAL CONTAINERS NO": { "type": "integer" },
    "20FT CONTAINERS NO": { "type": "integer" },
    "40FT CONTAINERS NO": { "type": "integer" },
    "INVOICE NO": { "type": "string" },
    "TOTAL AMOUNT": { "type": "number" },
    "CARRIER COMPANY": { "type": "string" },
    "BILL OF LADING NO": { "type": "string" }
  },
```

```
  "required": [
    "OCEAN VESSEL NAME",
    "PORT OF LOADING",
    "INVOICE DATE",
    "SHIPPED ON BOARD DATE",
    "QUANTITY",
    "PACKING",
    "TOTAL CONTAINERS NO",
    "20FT CONTAINERS NO",
    "40FT CONTAINERS NO",
    "INVOICE NO",
    "TOTAL AMOUNT",
    "CARRIER COMPANY",
    "BILL OF LADING NO"
  ]
}
"""
```

(**b**)

Figure 2. Prompt for the zero-shot settings (**a**) and prompt contents, and (**b**) schema data fields and requirement specifications.



```
INFORMATION_EXTRACTION_PROMPT = ("A content is shown below.\n\n"
    "- Please understand the below context and format the output as a JSON instance that conforms to the JSON schema below.\n\n"
    "- If you cannot find the correspond part inside the context, answer it with null.\n\n"
    "Here is the output schema:\n"
    "{schema}\n\n"
    "# Example Output:\n\n"
    "{example_output}\n\n"
    "---BEGIN CONTEXT---\n\n"
    "{response_text}\n\n"
    "---END CONTEXT---"
    )
```

(a)

```
OUTPUT="""{
  "OCEAN VESSEL NAME": "NAME OF VESSEL",
  "PORT OF LOADING": "NAME OF PORT",
  "INVOICE DATE": "2023-02-03",
  "SHIPPED ON BOARD DATE": "2023-02-05",
  "QUANTITY": 123.45,
  "PACKING": "12 X 40' CONTAINERS",
  "TOTAL CONTAINERS NO": 12,
  "20FT CONTAINERS NO": 0,
  "40FT CONTAINERS NO": 12,
  "INVOICE NO": "S-12345A",
  "TOTAL AMOUNT": 12345.67,
  "CARRIER COMPANY": "NAME OF COMPANY",
  "BILL OF LADING NO": "ABC123456789"
}
```

(**b**)

```
{
  "OCEAN VESSEL NAME": "NAME OF VESSEL2",
  "PORT OF LOADING": "NAME OF PORT2",
  "INVOICE DATE": "2023-05-01",
  "SHIPPED ON BOARD DATE": "2023-05-25",
  "QUANTITY": 123.456,
  "PACKING": "5 X 20' CONTAINERS",
  "TOTAL CONTAINERS NO": 5,
  "20FT CONTAINERS NO": 20,
  "40FT CONTAINERS NO": 0,
  "INVOICE NO": "123456789",
  "TOTAL AMOUNT": 12.3,
  "CARRIER COMPANY": "NAME OF COMPANY2",
  "BILL OF LADING NO": "123456789"
}
```

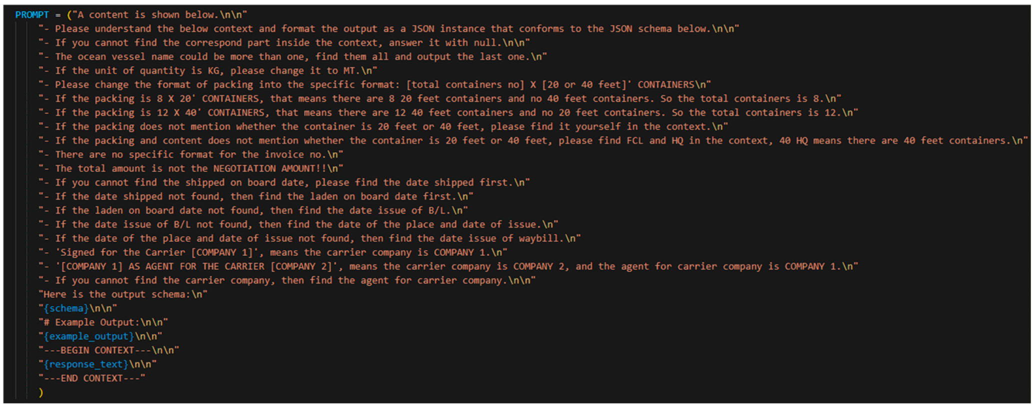
(**c**)

```
{
  "OCEAN VESSEL NAME": "NAME OF VESSEL3",
  "PORT OF LOADING": "NAME OF PORT3",
  "INVOICE DATE": "2024-08-10",
  "SHIPPED ON BOARD DATE": "2024-08-21",
  "QUANTITY": 1234.5,
  "PACKING": "7 X 40' CONTAINERS",
  "TOTAL CONTAINERS NO": 7,
  "20FT CONTAINERS NO": 0,
  "40FT CONTAINERS NO": 40,
  "INVOICE NO": "12345-A",
  "TOTAL AMOUNT": 1234.567,
  "CARRIER COMPANY": "NAME OF COMPANY3",
  "BILL OF LADING NO": "ABCD123456789"
}
"""
```

(**d**)

Figure 3. Prompt settings for (**a**) extraction schema (**b**) one-shot learning, (**c**,**d**) few-shot learning.

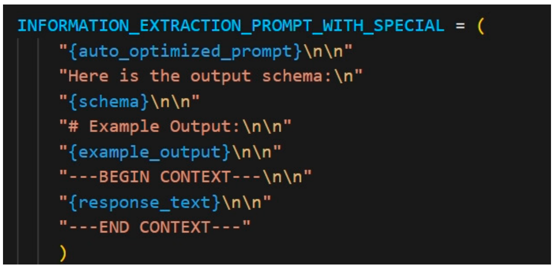
In Figure 4, the content of the manual prompt is the set of requirements, conditions, and instructions. As can be seen from the figure, a detailed set of processing instructions is provided for each field. Thus, even though the GPT-3.5 Turbo can automate the prompt processing, a detailed instruction set is provided for the prompt-based field value processing that can overcome the challenges and performance issues. These instructions are the specific requirements of the target company that provided the initial software requirements for document processing by the LLM.



```
PROMPT = ("A content is shown below.\n\n"
    "- Please understand the below context and format the output as a JSON instance that conforms to the JSON schema below.\n\n"
    "- If you cannot find the correspond part inside the context, answer it with null.\n\n"
    "- The ocean vessel name could be more than one, find them all and output the last one.\n"
    "- If the unit of quantity is KG, please change it to MT.\n"
    "- Please change the format of packing into the specific format: [total containers no] X [20 or 40 feet]' CONTAINERS\n"
    "- If the packing is 8 X 20' CONTAINERS, that means there are 8 20 feet containers and no 40 feet containers. So the total containers is 8.\n"
    "- If the packing is 12 X 40' CONTAINERS, that means there are 12 40 feet containers and no 20 feet containers. So the total containers is 12.\n"
    "- If the packing does not mention whether the container is 20 feet or 40 feet, please find it yourself in the context.\n"
    "- If the packing and content does not mention whether the container is 20 feet or 40 feet, please find FCL and HQ in the context, 40 HQ means there are 40 feet containers.\n"
    "- There are no specific format for the invoice no.\n"
    "- The total amount is not the NEGOTIATION AMOUNT!!\n"
    "- If you cannot find the shipped on board date, please find the date shipped first.\n"
    "- If the date shipped not found, then find the laden on board date first.\n"
    "- If the laden on board date not found, then find the date issue of B/L.\n"
    "- If the date issue of B/L not found, then find the date of the place and date of issue.\n"
    "- If the date of the place and date of issue not found, then find the date issue of waybill.\n"
    "- 'Signed for the Carrier [COMPANY 1]', means the carrier company is COMPANY 1.\n"
    "- '[COMPANY 1] AS AGENT FOR THE CARRIER [COMPANY 2]', means the carrier company is COMPANY 2, and the agent for carrier company is COMPANY 1.\n"
    "- If you cannot find the carrier company, then find the agent for carrier company.\n\n"
    "Here is the output schema:\n"
    "{schema}\n\n"
    "# Example Output:\n\n"
    "{example_output}\n\n"
    "---BEGIN CONTEXT---\n\n"
    "{response_text}\n\n"
    "---END CONTEXT---"
    )
```

Figure 4. Manual prompt contents.

In Figure 5, the auto-prompt designed as exclusive applied KIE contains few-shot training, which is generated from the APE [44] as synthetic cases. Users' manually annotated cases are then added as field definitions and the output schema is specified with three examples of output similar to the few-shot prompt. Subsequently, the LLM generates a new prompt based on the above settings. Next, the auto_optimized_prompt tag is then used to optimize the prompt and this process is repeated until an optimal point is reached and output is produced. The optimization used here is to improve model performance and provide effective results. Finally, the response text is presented with the respective queries asked regarding the output enquiry. The tasks have generation and classification as the same method, which contains the initial prompt and task description to be filled, and the result is the optimized prompt output.



```
INFORMATION_EXTRACTION_PROMPT_WITH_SPECIAL = (
    "{auto_optimized_prompt}\n\n"
    "Here is the output schema:\n"
    "{schema}\n\n"
    "# Example Output:\n\n"
    "{example_output}\n\n"
    "---BEGIN CONTEXT---\n\n"
    "{response_text}\n\n"
    "---END CONTEXT---"
    )
```

Figure 5. Applied KIE prompt contents with intent-based prompt calibration.

3.4. Experiments

The system configuration setup in Table 1 for the applied KIE implementation and experiments is as follows:

Table 1. System configuration.

Item	Description	Manufacturer	City and Country
System	Workstation (Windows 10, 64-bit OS)	Microsoft	Redmond, DC, USA
Processor	Intel(R) Core(TM) i7-8565U CPU @ 1.80 GHz	Intel Corporation	Santa Clara, CA, USA
Memory	Kingston 16 GB, SODIMM	Kingston Technology	Fountain Valley, CA, USA

We experimented with two datasets to examine the feasibility of the prompt engineering approaches. One is public (SROIE Dataset) and another is derived from receipts issued by a Taiwanese shipping company. The SROIE consists of a dataset with 1000 whole scanned receipt images and annotations for the competition on scanned receipts' OCR and key information extraction [45]. Table 2 presents the shipping company data provided by the Taiwanese shipping company. The details include 68 different types of invoices format, lading bills, the number of shipping companies that are involved within the import–export of containers, average document pages, and document size. It is necessary to know about the previous practices, so that the system user can interact with the LLM capturing invoice data and provide query-based answers. The APE-based LLM provides effective reasoning for invoice document analysis. Therefore, to provide accurate and detailed reasoning for the input query by the user, GPT-3.5 Turbo and GPT-4o are utilized with a high number of token capacity and low temperature parameters to ensure output consistency, as shown in Table 3.

Table 2. Details of shipping company data.

Number of Invoices	Lading Bills Formats	Shipping Company Formats	Average Document Contents (Pages)	Average Document Size
68	19	17	4	481.08 KB

Table 3. Large language models' parameters.

Model Name	Context Window Size	Temperature Parameter
GPT-3.5-Turbo	16,385 Tokens	0
GPT-4o	128,000 Tokens	0

The parameters in Table 4 are a necessary part of the GPT settings for fine-tuning performance. The K value specifying "How much ratings can change?" is used to restrict the token selection for the most likely options. The parameters are controlled by the adjustment magnitude using the Elo rating system. We tested different settings of NUMBER_OF_PROMPTS to observe the scaling behavior of APE. The results demonstrated that increasing the number of prompt candidates beyond three offered only a marginal improvement in extraction accuracy (less than 2%), while computational costs increased proportionally. Given the goal of providing a low-cost, practical KIE pipeline, we set NUMBER_OF_PROMPTS = 3 as the default configuration, which balances extraction performance and efficiency.

$$R_A + K * (Score_A - E_A)$$

$$R_B + K * ((1 - Score_A) - E_B)$$

where R is the rating and E is the expected scores for zero-sum games between **A** and **B**. Through this formula, K controls the update magnitude of the score. For example, with a K value of 32, the maximum change per update is 32 points. The candidate model for the evaluation version is GPT-4, as it should be higher than the training model, whereas the generation model and ranking model are "GPT-3.5-Turbo", as discussed earlier. It is used to generate multiple candidate prompts using a candidate model then using a generative model to generate output text based on each prompt. Later, score and rank tips are compared with the generated output text using a ranking model. The candidate model temperature is 0.9, which means randomness of the generated responses. A higher value as

0.9 makes responses more diverse. Similarly, for the generation, model temperature is kept higher at 0.8. Whereas the ranking temperature is 0.5 as medium meaning, consisting of mixed diverse and deterministic responses. The generation model max tokens are assigned as 60, indicating the generated response length limit. Appropriate value control fits desired context and the response length. Ultimately, limited tokens ensure control of cost and response time. Nevertheless, the number of prompts determines "How many candidates prompts to generate?". The higher, the more expensive, but the better the results will be.

Table 4. GPT prompt model parameters.

Model Parameters	Value
K	32
CANDIDATE_MODEL	'GPT-4'
CANDIDATE_MODEL_TEMPERATURE	0.9
GENERATION_MODEL	'GPT-3.5-Turbo'
GENERATION_MODEL_TEMPERATURE	0.8
GENERATION_MODEL_MAX_TOKENS	60
RANKING_MODEL	'GPT-3.5-Turbo'
RANKING_MODEL_TEMPERATURE	0.5
NUMBER_OF_PROMPTS	3

3.5. Contents and Setup of Baseline Queries

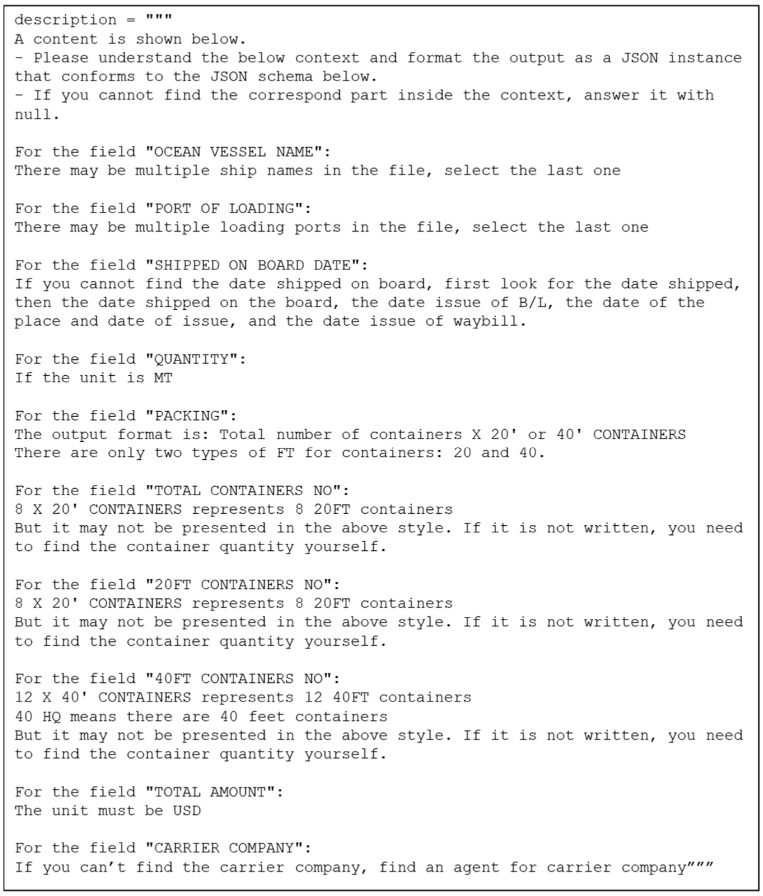
APE-There will be an initial description and several test cases to fill in, with the result having optimized description. To achieve it, the prompt test cases include the following instructions:

a. "Please help me analyze these invoices and bills of lading".
b. "Please find the target fields of these invoices and bills of lading".
c. "Please extract the target fields in these invoices and bills of lading".

IPC-There will be an initial prompt and task description to complete, and the result will be the optimization of the initial prompt. The ranking for the label scheme is ["1", "2", "3", "4", "5"] with a record value of 50. Here, the predictor is GPT-4-1106-preview and the temperature is set to 0.8.

For the prompt given in Figure 6, the task description is given as "Please extract the target fields from these invoices and bills of lading". The input queries' constraints given to the applied KIE prompt for processing contain the following statements, as given in Figure 6. The input context details given to the GPT prompt engineer are shown in Figure 2. The details regarding the field definition settings for the invoices are presented for the GPT initiation.

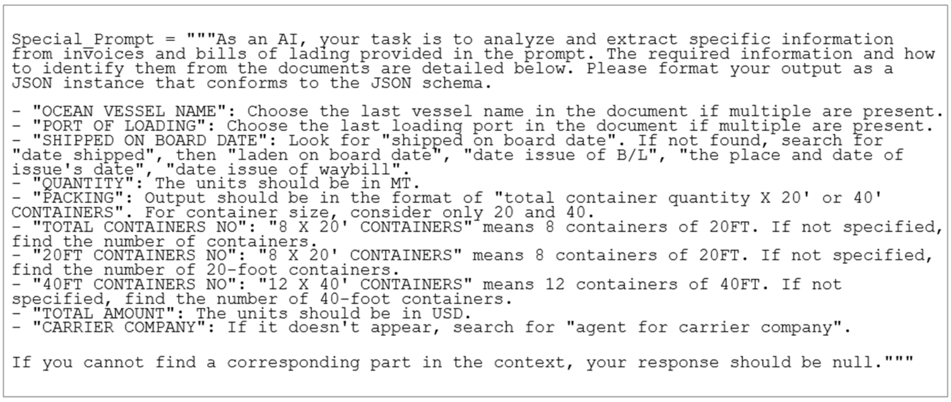
The details are presented to the GPT for understanding the JSON schema processing and later, the data can be used by the prompt for reasoning the user queries. A total of 10 field details for processing are provided that help the prompt to understand, from which the JSON schema structure determines if the data should be picked up and with what field measure/unit. Additional details regarding the field contents are also provided to understand the coding scheme, e.g., 8 × 20 ft means 8 containers that are 20 feet in size. The above fields are the practical implementation details that are asked by the company engineers to be provided in the applied KIE system. This context is mostly related to the enquiry of the import/export of the containers by shipping route. Every container may be different based on the date of transport, invoice number, container size, ocean vessel name, loading port, shipping date, carrier company, total cost, lading number, etc. Even though the enquiry data can be read easily, if the bills contain many pages and need to be kept safely for the storage purpose, then once scanned, it can reveal many details for this AI assistant-based system model.



```
description = """
A content is shown below.
- Please understand the below context and format the output as a JSON instance
that conforms to the JSON schema below.
- If you cannot find the correspond part inside the context, answer it with
null.

For the field "OCEAN VESSEL NAME":
There may be multiple ship names in the file, select the last one

For the field "PORT OF LOADING":
There may be multiple loading ports in the file, select the last one

For the field "SHIPPED ON BOARD DATE":
If you cannot find the date shipped on board, first look for the date shipped,
then the date shipped on the board, the date issue of B/L, the date of the
place and date of issue, and the date issue of waybill.

For the field "QUANTITY":
If the unit is MT

For the field "PACKING":
The output format is: Total number of containers X 20' or 40' CONTAINERS
There are only two types of FT for containers: 20 and 40.

For the field "TOTAL CONTAINERS NO":
8 X 20' CONTAINERS represents 8 20FT containers
But it may not be presented in the above style. If it is not written, you need
to find the container quantity yourself.

For the field "20FT CONTAINERS NO":
8 X 20' CONTAINERS represents 8 20FT containers
But it may not be presented in the above style. If it is not written, you need
to find the container quantity yourself.

For the field "40FT CONTAINERS NO":
12 X 40' CONTAINERS represents 12 40FT containers
40 HQ means there are 40 feet containers
But it may not be presented in the above style. If it is not written, you need
to find the container quantity yourself.

For the field "TOTAL AMOUNT":
The unit must be USD

For the field "CARRIER COMPANY":
If you can't find the carrier company, find an agent for carrier company”””
```

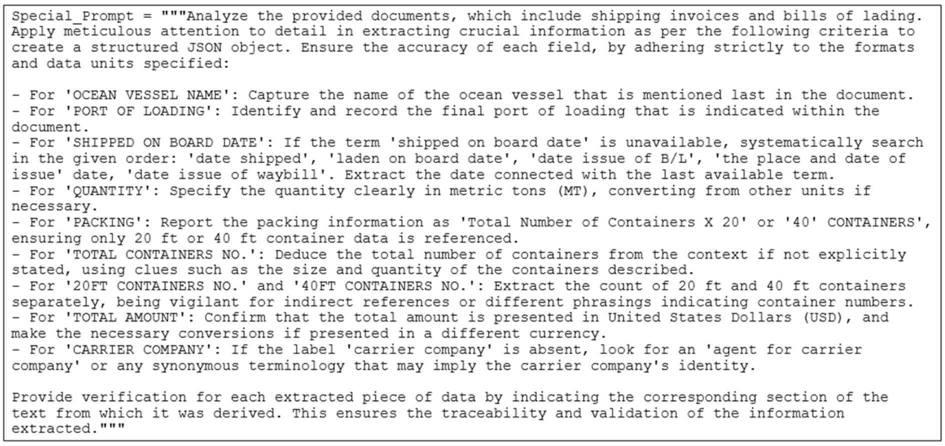
Figure 6. GPT prompt engineer input for field definition settings of invoices.

The above prompt contents for the field definitions are optimized for the applied KIE algorithm during the optimization. The optimization process is iterated until the stopping criteria for the best performance are reached. As shown in Figure 7, the field definitions are improved as per the APE algorithm [14]. Next, Figure 8 shows the starting and ending instructions for the LLM prompt to perform improved processing, generated as per the IPC algorithm [44].



```
Special_Prompt = """As an AI, your task is to analyze and extract specific information
from invoices and bills of lading provided in the prompt. The required information and how
to identify them from the documents are detailed below. Please format your output as a
JSON instance that conforms to the JSON schema.

- "OCEAN VESSEL NAME": Choose the last vessel name in the document if multiple are present.
- "PORT OF LOADING": Choose the last loading port in the document if multiple are present.
- "SHIPPED ON BOARD DATE": Look for "shipped on board date". If not found, search for
"date shipped", then "laden on board date", "date issue of B/L", "the place and date of
issue's date", "date issue of waybill".
- "QUANTITY": The units should be in MT.
- "PACKING": Output should be in the format of "total container quantity X 20' or 40'
CONTAINERS". For container size, consider only 20 and 40.
- "TOTAL CONTAINERS NO": "8 X 20' CONTAINERS" means 8 containers of 20FT. If not specified,
find the number of containers.
- "20FT CONTAINERS NO": "8 X 20' CONTAINERS" means 8 containers of 20FT. If not specified,
find the number of 20-foot containers.
- "40FT CONTAINERS NO": "12 X 40' CONTAINERS" means 12 containers of 40FT. If not
specified, find the number of 40-foot containers.
- "TOTAL AMOUNT": The units should be in USD.
- "CARRIER COMPANY": If it doesn't appear, search for "agent for carrier company".

If you cannot find a corresponding part in the context, your response should be null."""
```

Figure 7. Output prompt configuration by APE.



```
Special_Prompt = """Analyze the provided documents, which include shipping invoices and bills of lading.
Apply meticulous attention to detail in extracting crucial information as per the following criteria to
create a structured JSON object. Ensure the accuracy of each field, by adhering strictly to the formats
and data units specified:

- For 'OCEAN VESSEL NAME': Capture the name of the ocean vessel that is mentioned last in the document.
- For 'PORT OF LOADING': Identify and record the final port of loading that is indicated within the
document.
- For 'SHIPPED ON BOARD DATE': If the term 'shipped on board date' is unavailable, systematically search
in the given order: 'date shipped', 'laden on board date', 'date issue of B/L', 'the place and date of
issue' date, 'date issue of waybill'. Extract the date connected with the last available term.
- For 'QUANTITY': Specify the quantity clearly in metric tons (MT), converting from other units if
necessary.
- For 'PACKING': Report the packing information as 'Total Number of Containers X 20' or '40' CONTAINERS',
ensuring only 20 ft or 40 ft container data is referenced.
- For 'TOTAL CONTAINERS NO.': Deduce the total number of containers from the context if not explicitly
stated, using clues such as the size and quantity of the containers described.
- For '20FT CONTAINERS NO.' and '40FT CONTAINERS NO.': Extract the count of 20 ft and 40 ft containers
separately, being vigilant for indirect references or different phrasings indicating container numbers.
- For 'TOTAL AMOUNT': Confirm that the total amount is presented in United States Dollars (USD), and
make the necessary conversions if presented in a different currency.
- For 'CARRIER COMPANY': If the label 'carrier company' is absent, look for an 'agent for carrier
company' or any synonymous terminology that may imply the carrier company's identity.

Provide verification for each extracted piece of data by indicating the corresponding section of the
text from which it was derived. This ensures the traceability and validation of the information
extracted."""
```

Figure 8. Output prompt configuration by IPC.

3.6. Performance Evaluation

For each entity type, if there is a matching predicted entity, it will be counted as a true positive (TP), otherwise it will be counted as a false negative (FN). Here, two entity mentions are considered a match if they have at least 80% string similarity between them. All remaining unmatched predicted entity mentions are considered false positives (FP) [46].

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2.P.R}{P + R} \tag{10}$$

The method of similarity calculation chosen for this study is fuzzy matching. Given the nature of the existing dataset, the "fuzz.token_set_ratio (TSeR)" function of the fuzzy wuzzy package was used.

This method considers partial matching of words and the concept of sets, meaning it compares partial content of two strings without requiring an exact match. It is suitable for handling strings with different lengths, word orders, but containing similar segments. Since there are abbreviation issues with carrier companies, a mapping between company names and their abbreviations will be provided. Before evaluating similarity, preprocessing will be carried out to unify them into abbreviations. Additionally, preprocessing will also remove the word "PORT" from the port of loading to increase the accuracy of the study.

4. Results and Discussion

4.1. Prompt Engineering Performance Comparison on SROIE Datasets

As illustrated in Table 5, a comparative analysis of the results derived from the SROIE dataset is presented. An experiment was conducted to ascertain the efficacy of key information extraction in the datasets using GPT-3.5-turbo. This experiment involved the zero-shot, one-shot, and few-shot methods. The overall accuracy, measured as "Document-Level Extraction Accuracy," exhibits a slight improvement from zero-shot (0.912) to few-shot (0.915). Although the discrepancy is negligible, it underscores the significance of expeditious optimization for precision.

Table 5. Results of performance comparison in SROIE datasets.

	Zero-Shot			One-Shot			Few-Shot		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Address	0.99	0.76	0.86	0.99	0.69	0.81	0.99	0.73	0.84
Company	0.85	0.89	0.87	0.85	0.85	0.85	0.88	0.89	0.88
Date	0.99	0.93	0.96	0.99	0.92	0.96	1.00	0.94	0.97
Total	0.97	0.95	0.96	0.92	0.83	0.87	0.95	0.92	0.93
Document Extraction Accuracy		0.912			0.912			0.915	

Note: Document Extraction Accuracy: A document is considered correct only if all required fields are accurately extracted.

4.2. Prompt Engineering Performance Comparison on Shipping Company Data

Tables 6–8 show a comparison of the results of different configured prompts with GPT-3.5-Turbo (Tables 6 and 7) and GPT-4o (Table 8). A total of 13 fields extracted from the carrier's invoices are taken as features and the different prompt comparisons are compared with the precision, recall, and F1-score for the same. The configured prompts with GPT-4o seem to have performed marginal improvement in document information extraction. The results showed that better scores are present in the applied KIE based on APE and manual prompt. Therefore, as most of the fields with the better recall, precision, and F1-scores are found in the applied KIE based on APE, it is selected as the top scorer with minimal errors in data interpretation with the LLM.

Table 6. Performance comparison for configured prompt with GPT-3.5-Turbo.

	Zero-Shot			One-Shot			Few-Shot		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Ocean Vessel Name	0.93	1	0.96	0.90	1	0.95	0.87	1	0.93
Port of Loading	0.96	1	0.98	0.96	1	0.98	0.96	1	0.98
Invoice Date	0.87	0.87	0.87	0.93	0.93	0.93	0.88	0.88	0.88
Shipped on Board Date	0.87	0.87	0.87	0.94	0.94	0.94	0.90	0.90	0.90
Quantity	0.91	0.91	0.91	0.94	0.94	0.94	0.96	0.96	0.96
Invoice No	0.99	0.99	0.99	0.97	0.97	0.97	0.97	0.97	0.97
Total Amount	1	1	1	0.99	0.99	0.99	0.96	0.96	0.96
Carrier Company	0.8	1	0.89	0.76	1	0.86	0.67	1	0.80
Bill of Lading No	0.98	1	0.99	0.98	1	0.99	0.98	1	0.99
Total Containers No	0.99	0.99	0.99	1	1	1	0.99	0.99	0.99

Table 6. *Cont.*

	Zero-Shot			One-Shot			Few-Shot		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
20ft Containers No	0.74	0.74	0.74	0.53	0.79	0.63	0.88	0.88	0.88
40ft Containers No	0.44	0.44	0.44	0.72	0.59	0.65	0.81	0.81	0.81
Packing	0.12	1	0.21	0.59	1	0.74	0.88	1	0.94
Document Extraction Accuracy	0.118			0.324			0.500		

Note: Document Extraction Accuracy: A document is considered correct only if all required fields are accurately extracted.

Table 7. Performance comparison for APE using GPT-3.5-Turbo with configured prompt.

	Applied KIE Based on APE			IPC			Manual Prompt			Human
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**	**Recall**
Ocean Vessel Name	0.95	1	0.97	0.98	1	0.99	0.98	1	0.99	1
Port of Loading	0.96	1	0.98	0.96	1	0.98	0.95	1	0.97	0.88
Invoice Date	1	1	1	0.99	0.99	0.99	0.96	0.96	0.96	0.97
Shipped on Board Date	0.97	0.97	0.97	0.99	0.99	0.99	0.99	0.99	0.99	0.93
Quantity	1	1	1	1	1	1	1	1	1	1
Invoice No	0.99	0.99	0.99	0.91	0.91	0.91	0.97	0.97	0.97	0.88
Total Amount	1	1	1	0.97	0.97	0.97	1	1	1	1
Carrier Company	0.80	1	0.89	0.81	1	0.90	0.95	1	0.97	0.99
Bill of Lading No	0.98	1	0.99	0.98	1	0.99	0.95	1	0.97	0.97
Total Containers No	1.	1	1	1	1	1	1	1	1	1
20ft Containers No	0.99	1	0.99	0.94	1	0.97	1	1	1	1
40ft Containers No	0.99	1	0.99	0.94	1	0.97	1	0.97	0.98	1
Packing	1	1	1	1	1	1	0.98	1	0.99	1
Document Extraction Accuracy	0.868			0.779			0.853			0.706

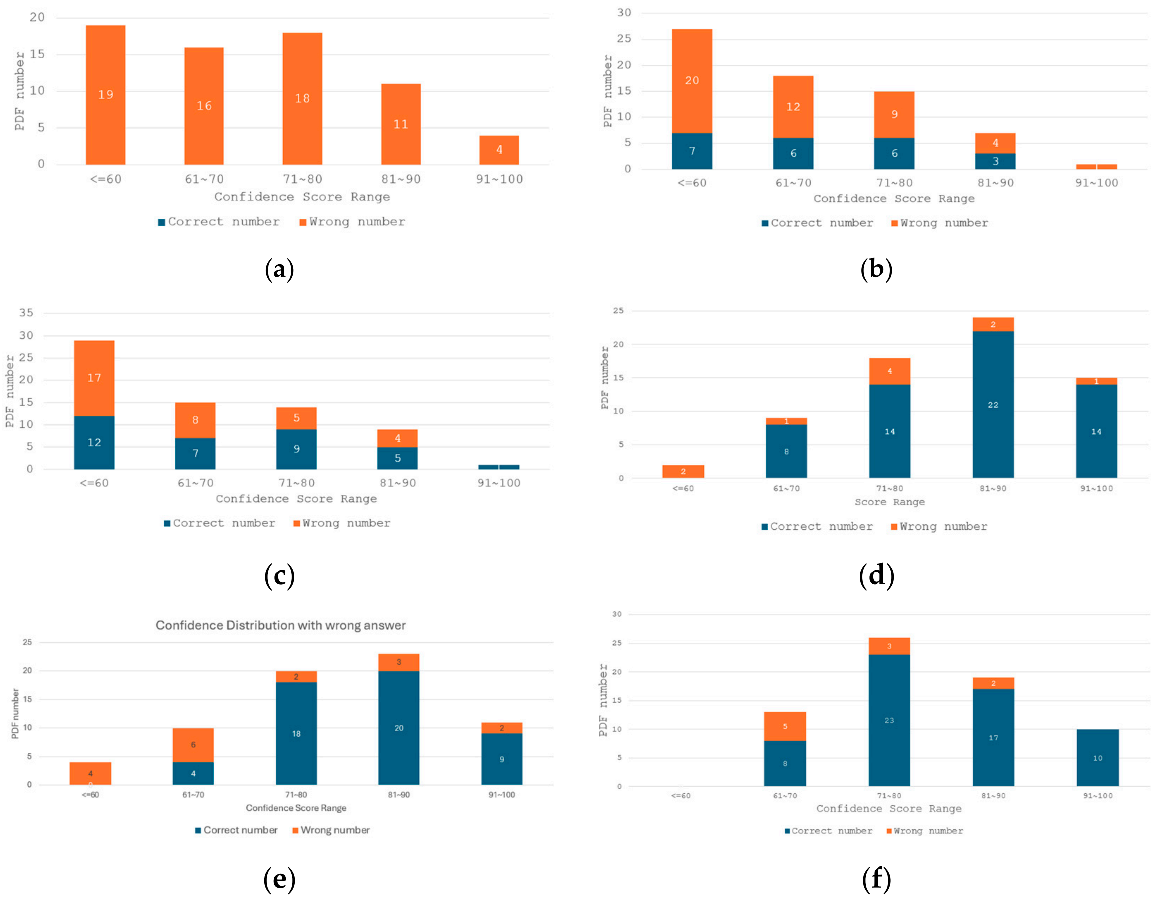
Note: Document Extraction Accuracy: A document is considered correct only if all required fields are accurately extracted.

Table 8. Performance comparison for APE using GPT-4o with configured prompt.

	Applied KIE Based on APE			IPC		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Ocean Vessel Name	0.98	1	0.99	0.96	1	0.98
Port of Loading	1	1	1	0.95	1	0.98
Invoice Date	1	1	1	0.99	0.99	0.99
Shipped on Board Date	0.97	0.97	0.97	0.97	0.97	0.97
Quantity	1	1	1	1	1	1
Invoice No	0.96	0.96	0.96	0.94	0.94	0.94
Total Amount	1	1	1	1	1	1
Carrier Company	0.78	1	0.88	0.78	1	0.88
Bill of Lading No	0.98	1	0.99	0.98	1	0.99
Total Containers No	1	1	1	1	1	1
20ft Containers No	1	1	1	1	1	1
40ft Containers No	1	1	1	1	1	1
Packing	1	1	1	1	1	1
Document Extraction Accuracy		0.882			0.824	

Note: Document Extraction Accuracy: A document is considered correct only if all required fields are accurately extracted.

Figure 9 shows the confidence scores for different prompt-based learning using GPT-3.5-Turbo. Figure 9a shows the zero-shot confidence score for the 20 pdf data collection. There is a high number of errors during data processing with zero-shot learning because the required output content types and conversions are unknown. In Figure 9b, the confidence scores based on one-shot learning show a slight improvement in the response as an example of the expected response is fed into the system. Similarly, Figure 9c shows the average improvements over the full confidence analysis by the few-shot learning, which consists of three examples of learning as the default output.



(**a**)

(**b**)

(**c**)

(**d**)

(**e**)

(**f**)

Figure 9. Confidence distribution: (**a**) zero-shot, (**b**) one-shot, (**c**) few-shot, (**d**) manual prompt, (**e**) IPC, and (**f**) applied KIE prompt based on APE.

Even a single error in document processing can be costly, as the data required from previous shipping orders, waybills, and bills of lading may not be acceptable. Figure 9d shows the improved confidence scores from the manual prompt, which has fully defined instructions set by the company itself. However, some errors seen in the score ≤ 60 are due to the inability to find dates for two queries, e.g., the required date format is 2023-10-11, but the generated example shows 2023-11-10. Additionally, the last error found in the score 91 to 100 is due to the OCR form analysis error. Figure 9e shows the IPC prompt with the fine-tuned confidence scores. The IPC consists of a human annotator and an LLM annotator for the classification pipeline. Although the results obtained are quite improved compared to the previous comparisons, it still has another window for high scores. Next, Figure 9f shows the applied KIE, which consists of generation and classification tasks together. In this case, the initial prompt and task description are added to obtain the optimal prompt later. The confidence score shows the best scores achieved.

We further explored the baseline QA and DQA models to experiment with the problems that would occur under the same task. The QA model is used by roberta-base model, fine-tuned using the SQuAD2.0 dataset. The DQA model is used by DocQuery. By comparing Tables 7 and 8 (proposed method) and Table 9 (baseline models), we observed that our method achieves significantly higher performance across key metrics, including precision recall, F1-score, and Document Extraction Accuracy. These results demonstrate the adaptability and scalability of our approach for handling diverse and complex document types.

Table 9. Comparison results for QA and DQA models.

	QA			DQA		
	Precision	**Recall**	**F1-Score**	**Precision**	**Recall**	**F1-Score**
Ocean Vessel Name	0.27	0.21	0.23	0.18	0.15	0.16
Port of Loading	0.35	0.24	0.28	0.34	0.19	0.25
Invoice Date	0.01	0.01	0.01	0.03	0.03	0.03
Shipped on Board Date	0.09	0.09	0.09	0.06	0.06	0.06
Quantity	0.62	0.62	0.62	0.62	0.62	0.62
Invoice No	0.59	0.59	0.59	0.87	0.87	0.87
Total Amount	0.24	0.24	0.24	0.44	0.44	0.44
Carrier Company	0	0	0	0.20	0.12	0.15
Bill of Lading No	0.27	0.24	0.25	0.55	0.44	0.49
Total Containers No	0.24	0.24	0.24	0.31	0.31	0.31
20ft Containers No	0.07	0.07	0.07	0.09	0.09	0.09
40ft Containers No	0.13	0.13	0.13	0.13	0.13	0.13
Packing	0.07	0.06	0.07	0.06	0.04	0.05
Document Extraction Accuracy		0.000			0.000	

Note: Document Extraction Accuracy: A document is considered correct only if all required fields are accurately extracted.

4.3. Analysis of Types of Errors

We proceeded to investigate the types of error by employing various approaches. When each field is considered in isolation, the results obtained by human staff members are consistently accurate. However, when a file is examined, if an error is identified, the staff's accuracy is reduced to approximately 70%. It is evident that the LLM, when provided solely with the schema to retrieve the fields, is unable to achieve the objective. However, when presented with a single example (one-shot), there is a 32.4% enhancement in performance. This improvement is further augmented to 50% when the LLM is exposed to three examples (few-shot). Nevertheless, the LLM continues to demonstrate low performance on the 20FT CONTAINERS NO and 40FT CONTAINERS NO reasoning.

However, upon optimization of the user requirements by the prompt, the score attains comparability to the prompt written by the AI engineers (manual prompt) and attains a higher value in certain fields. It is evident that APE has obtained a better score of 86.8% with GPT-3.5 turbo and 88.2% with GPT-4o, while IPC has attained 77.9% with GPT-3.5

turbo and 82.4% with GPT-4o. This disparity can be attributed to the necessity of manual scoring of data in the IPC generation method, which may not align as closely with the consistency of the APE LLM evaluation standard.

Based on the experimental results and manual inspection, we have organized the main types of extraction errors in our experiments. Table 10 summarizes the main types of errors encountered, current solutions, and possible future improvement strategies.

Table 10. Error type summary.

Error Type	Description	Current Handling	Future Mitigation Plan
Date Format Error	Generated wrong date formats (e.g., required 2023-10-11 but generated 2023-11-10).	Manual prompt optimization improved date formatting.	Further enhance prompt clarity and stricter validation for date fields.
Reasoning Error: 20FT CONTAINERS NO/40FT CONTAINERS NO	LLM struggles to reason about container counts correctly.	Manual prompt optimization improved, but errors remain.	Apply post-processing (e.g., rule-based count correction); explore layout-based hints.
OCR Form Analysis Error	Some errors from incorrect OCR extraction (e.g., missing or misaligned fields).	Confidence score thresholding to detect suspicious outputs.	Evaluate alternative OCR tools (e.g., Google Document AI) or apply OCR correction.

4.4. Discussion

APE Performance: Designing an optimal prompt for the commercial application will require a complete set of rules defined in detail for every field and a tractable iterative function for improvement. Therefore, an efficiently designed prompt dominates the other basic prompt performance. The confidence score achieved is higher for the prompt-based response for every user-specific query expecting reasoning. Successively, the precision score for the different field-based evaluations is also highest for having fewer mistakes.

Manual Prompt Configuration: A human designed prompt was utilized in this experiment. Instead of asking the LLM to retrieve data from the pdf, the manual prompt is constructed with a detailed description for every field datum and value capturing method. Therefore, the performance achieved by the manual prompt was in the top three precision and confidence scores. Even though the high score is impressive, it would have limitations on longer written prompts including manual testing and might require updating the prompt contents with the detailed description for the change in fields, tax, etc.

Quality Control Measure: A quality parameter needs to be set for measuring the model's effectiveness. While evaluating different field data, every field should produce correct values, e.g., lading, billing amount, tax, etc. Thus, the confidence score can indicate whether the particular response is completely or partially trustable. Henceforth, having a high confidence is crucial for the model to be used in commercial applications.

Applied KIE Compatibility and Applicability: Prompt usage reduces the system model training requirements for large datasets as required for deep learning in comparison. Therefore, the use of easy to understand rules with few-shot learning can provide the best performance. Also, the APE framework-based system can help to provide multiple use cases to test the prompt fitness applicability. Even though the LLM prompt is upgraded, the changes required in the system are minimal for versioning. Nevertheless, the model design and configurations for the prompt required are less compared to the traditional/deep learning systems. Human-assisted optimization is performed in the algorithm by iterating over conditions until performance improvement.

Applied KIE Scalability and Generalization: While our primary focus is on industrial invoices and shipping documents in English, the proposed system architecture is inherently modular and can be adapted to other document types and languages. First, given Amazon Textract's and similar OCR engines' ability to extract text from a variety of document formats (e.g., contracts, certificates, reports), the system can be extended to other domains by simply updating the field schema definitions and hint templates. The hint engineering strategy enables domain experts to dynamically define new fields and relationships without requiring retraining of the LLM. Secondly, given the capacity of models such as GPT-3.5 Turbo to support multilingual input, the translation of cue templates and instructions into the target language becomes a possibility. Alternatively, a combination of OCR with language-specific preprocessing can be employed to expand the system's applicability to non-English documents. Future research could further explore the use of a multilingual LLM or fine-tuned cues to optimize performance on highly structured documents in languages such as Chinese, Japanese, or Spanish. These features underscore the extensive scalability potential of the proposed approach across various industries, file types, and programming languages, with only minor architectural modifications.

Ethical Considerations in Using LLMs for Sensitive Business Documents: The use of large language models (LLMs) in the processing of sensitive business documents raises several ethical challenges, particularly with respect to privacy, fairness, and the risk of bias. In this study, we acknowledge these concerns and have taken proactive measures to address them. First, to protect privacy, sensitive information is anonymized or masked before it is processed by the models, ensuring that no personal or confidential data are exposed. In addition, we focus on mitigating bias and ensuring fairness in the outputs generated by the LLMs. By using prompt engineering techniques, we ensure that the data provided to the models are consistent, reducing the risk of biased or discriminatory responses. This is particularly important in the context of business document processing, where fairness and impartiality are essential. These precautions are critical to ensuring the ethical use of LLMs, and we are committed to maintaining privacy, fairness, and transparency throughout the process.

Limitations for the Prompt Language Processing: Due to the language limitations, prompt processing for a multi-language document will differ. English is a standard communication language world-wide and easy to process due to prompt support as compared to some languages with four tones. Therefore, the training dataset will differ for the language dataset's availability and its completeness. Thus, adding the necessary language settings and data can make the prompt sustainable across all the countries' local language interactions. Input documents more than 16,385 tokens can be a major limitation for the GPT-3.5 turbo input processing.

5. Conclusions

The applied KIE is an effective approach for different categories of document processing using GPT in the industrial environment. Manual work for recording data from transactional receipts every day requires high accuracy and a considerable amount of time. Thus, data entry work is one of the major limitations faced by various industrial departments for their continuous operations and business dealings. Therefore, an IDP is necessary to ease the department's work and improve the functioning speed with accuracy.

Recently, the prompt engineer solution provided by the applied KIE is world-class for the industrial environment to be adopted as a quality standard. The applied KIE provides an active prompt for the user to interact with input documents and obtain detailed reasoning for the respective response. Therefore, the system is designed consisting of Amazon Textract for the data extraction by overcoming noise and the configuration module for the structured/unstructured data preprocessing. Prompt composition is made using rules by manual annotations, APE and IPC. Successively, GPT-3.5-Turbo and GPT-4o with multiple learning analysis were found to be efficient with few-shot settings by providing the best performance of 97% precision in comparison to other prompt settings without overhead. As the applied KIE provides efficient results across different transactional documents, its usage in the industrial environment is highly recommended. In future work, we would look forward to processing all categories of industrial documents including designs, images, and videos. Furthermore, the utilization of alternative large language models (LLMs), including DeepSeek, Claude, and domain-specific fine-tuned models, will be investigated for the purpose of evaluating their extraction accuracy, robustness, and cost-effectiveness. Similarly, we will explore different OCR engines, including Google Document AI and Azure Form Recognizer, to assess whether improved OCR quality can further improve the overall pipeline performance. These improvements may enable the system to handle an even broader range of document types and industry scenarios without requiring significant manual intervention.

Author Contributions: Conceptualization, L.-C.C. and R.-K.S.; methodology, L.-C.C. and K.-C.P.; validation, H.-T.W. and K.-C.P.; formal analysis, H.-T.W.; investigation, R.-K.S.; resources, R.-K.S.; data curation, R.-K.S.; writing—original draft preparation, L.-C.C., M.S.P., and K.-C.P.; writing—review and editing, L.-C.C.; visualization, H.-T.W.; supervision, C.-M.C.; project administration, R.-K.S. and K.-C.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the company provider on request.

Acknowledgments: We are grateful to DDS-THU AI Center for providing us with the required resources and problems to solve using Applied AI.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Avei, U.I.; Goularas, D.; Korkmaz, E.E.; Deveci, B. Information Extraction from Scanned Invoice Documents Using Deep Learning Methods. In Proceedings of the 2024 IEEE Thirteenth International Conference on Image Processing Theory, Tools and Applications (IPTA), Rabat, Morocco, 14–17 October 2024.
2. Khurana, D.; Koli, A.; Khatter, K.; Singh, S. Natural language processing: State of the art, current trends and challenges. *Multimed. Tools Appl.* **2023**, *82*, 3713–3744. [CrossRef]
3. Saout, T.; Lardeux, F.; Saubion, F. An Overview of Data Extraction From Invoices. *IEEE Access* **2024**, *12*, 19872–19886. [CrossRef]
4. Lin, W.; Gao, Q.; Sun, L.; Zhong, Z.; Hu, K.; Ren, Q.; Huo, Q. ViBERTgrid: A Jointly Trained Multi-Modal 2D Document Representation for Key Information Extraction from Documents. *arXiv* **2021**, arXiv:2105.11672.
5. Lam, L.; Ratnamogan, P.; Tang, J.; Vanhuffel, W.; Caspani, F. Information Extraction from Documents: Question Answering vs Token Classification in real-world setups. In Proceedings of the IEEE International Conference on Document Analysis and Recognition, San Jose, CA, USA, 21–26 August 2023.
6. Dong, Q.; Li, L.; Dai, D.; Zheng, C.; Wu, Z.; Chang, B.; Sun, X.; Xu, J.; Li, L.; Sui, Z. A Survey on In-context Learning. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 7–11 December 2022.
7. Yindumathi, K.M.; Chaudhari, S.S.; Aparna, R. Structured Data Extraction Using Machine Learning from Image of Unstructured Bills/Invoices. In *Smart Computing Techniques and Applications*; Satapathy, S.C., Bhateja, V., Favorskaya, M.N., Adilakshmi, T., Eds.; Smart Innovation, Systems and Technologies; Springer: Singapore, 2021; Volume 224.

8. Tang, Z.; Yang, Z.; Wang, G.; Fang, Y.; Liu, Y.; Zhu, C.; Zeng, M.; Zhang, C.; Bansal, M. Unifying Vision, Text, and Layout for Universal Document Processing. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
9. Kim, G.; Hong, T.; Yim, M.; Nam, J.; Park, J.; Yim, J.; Hwang, W.; Yun, S.; Han, D.; Park, S. OCR-Free Document Understanding Transformer. In Proceedings of the European Conference on Computer Vision, Montreal, QC, Canada, 11 October 2021.
10. Huang, Y.; Lv, T.; Cui, L.; Lu, Y.; Wei, F. LayoutLMv3: Pre-training for Document AI with Unified Text and Image Masking. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022.
11. Liu, W.; Shen, X.; Pun, C.; Cun, X. Explicit Visual Prompting for Low-Level Structure Segmentations. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
12. Liu, C.; Yin, K.; Cao, H.; Jiang, X.; Li, X.; Liu, Y.; Jiang, D.; Sun, X.; Xu, L. HRVDA: High-Resolution Visual Document Assistant. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 17–21 June 2024; pp. 15534–15545.
13. Appalaraju, S.; Tang, P.; Dong, Q.; Sankaran, N.; Zhou, Y.; Manmatha, R. DocFormerv2: Local Features for Document Understanding. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023.
14. Li, J.; Xu, Y.; Lv, T.; Cui, L.; Zhang, C.; Wei, F. DiT: Self-supervised Pre-training for Document Image Transformer. In Proceedings of the 30th ACM International Conference on Multimedia, Lisboa, Portugal, 10–14 October 2022.
15. Mahoney, C.J.; Zhang, J.; Huber-Fliflet, N.; Gronvall, P.; Zhao, H. A Framework for Explainable Text Classification in Legal Document Review. In Proceedings of the 2019 IEEE International Conference on Big Data, Los Angeles, CA, USA, 9–12 December 2019.
16. Ai, Q.; O'Connor, B.T.; Croft, W.B. A Neural Passage Model for Ad-hoc Document Retrieval. *arXiv* **2018**, arXiv:2103.09306.
17. Huang, Q.; Dong, X.; Chen, D.; Zhang, W.; Wang, F.; Hua, G.; Yu, N.H. Diversity-Aware Meta Visual Prompting. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
18. Sun, H.; Li, X.; Xu, Y.; Homma, Y.; Cao, Q.; Wu, M.; Jiao, J.; Charles, D.X. AutoHint: Automatic Prompt Optimization with Hint Generation. *arXiv* **2023**, arXiv:2307.07415.
19. Shum, K.; Diao, S.; Zhang, T. Automatic Prompt Augmentation and Selection with Chain-of-Thought from Labeled Data. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023.
20. Shin, T.; Razeghi, Y.; Logan, I.V.R.L.; Wallace, E.; Singh, S. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv* **2020**, arXiv:2010.15980.
21. Zhou, Y.; Muresanu, A.I.; Han, Z.; Paster, K.; Pitis, S.; Chan, H.; Ba, J. Large Language Models Are Human-Level Prompt Engineers. *arXiv* **2022**, arXiv:2211.01910.
22. Gao, T.; Fisch, A.; Chen, D. Making Pre-trained Language Models Better Few-shot Learners. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Bangkok, Thailand, 1–6 August 2021; Volume 1.
23. Wang, D.; Raman, N.; Sibue, M.; Ma, Z.; Babkin, P.; Kaur, S.; Pei, Y.; Nourbakhsh, A.; Liu, X. DocLLM: A layout-aware generative language model for multimodal document understanding. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023.
24. He, J.; Wang, L.; Hu, Y.; Liu, N.; Liu, H.; Xu, X.; Shen, H. ICL-D3IE: In-Context Learning with Diverse Demonstrations Updating for Document Information Extraction. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 December 2023.
25. Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; Zhou, M. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Toronto, ON, Canada, 3–7 August 2025.
26. Pryzant, R.; Iter, D.; Li, J.; Lee, Y.T.; Zhu, C.; Zeng, M. Automatic Prompt Optimization with "Gradient Descent" and Beam Search. In Proceedings of the Conference on Empirical Methods in Natural Language Processing, Singapore, 6–10 December 2023.
27. Ye, Q.; Axmed, M.; Pryzant, R.; Khani, F. Prompt Engineering a Prompt Engineer. In Proceedings of the Annual Meeting of the Association for Computational Linguistics, Toronto, ON, Canada, 9–14 July 2023.
28. Yang, C.; Wang, X.; Lu, Y.; Liu, H.; Le, Q.V.; Zhou, D.; Chen, X. Large Language Models as Optimizers. *arXiv* **2023**, arXiv:2309.03409.
29. Yu, Z.; Gao, T.; Zhang, Z.; Lin, Y.; Liu, Z.; Sun, M.; Zhou, J. Automatic Label Sequence Generation for Prompting Sequence-to-sequence Models. *arXiv* **2022**, arXiv:2209.09401.
30. Wang, Y.; Shen, S.; Lim, B.Y. RePrompt: Automatic Prompt Editing to Refine AI-Generative Art Towards Precise Expressions. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Hamburg, Germany, 23–28 April 2023.
31. Guo, L.; Wang, C.; Wang, X.; Zhu, L.; Yin, H. Automated Prompting for Non-overlapping Cross-domain Sequential Recommendation. *arXiv* **2023**, arXiv:2304.04218.

32. Oh, C.; Hwang, H.; Lee, H.; Lim, Y.; Jung, G.; Jung, J.; Choi, H.; Song, K. BlackVIP: Black-Box Visual Prompting for Robust Transfer Learning. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
33. Wang, N.; Xie, J.; Wu, J.; Jia, M.; Li, L. Controllable Image Captioning via Prompting. In Proceedings of the AAAI Conference on Artificial Intelligence, Online, 22 February–1 March 2022.
34. Mitra, C.; Huang, B.; Darrell, T.; Herzig, R. Compositional Chain-of-Thought Prompting for Large Multimodal Models. In Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 19–21 June 2024; pp. 14420–14431.
35. Chen, A.; Yao, Y.; Chen, P.; Zhang, Y.; Liu, S. Understanding and Improving Visual Prompting: A Label-Mapping Perspective. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
36. Wang, C.; Pan, J.; Wang, W.; Dong, J.; Wang, M.; Ju, Y.; Chen, J. PromptRestorer: A Prompting Image Restoration Method with Degradation Perception. In Proceedings of the Neural Information Processing Systems, New Orleans, LA, USA, 10–16 December 2023.
37. Zhang, X.; Gao, W. Towards LLM-based Fact Verification on News Claims with a Hierarchical Step-by-Step Prompting Method. In Proceedings of the International Joint Conference on Natural Language Processing, Bali, Indonesia, 1–4 November 2023.
38. Villa, A.; Alc'azar, J.L.; Alfarra, M.; Alhamoud, K.; Hurtado, J.; Heilbron, F.C.; Soto, Á.; Ghanem, B. PIVOT: Prompting for Video Continual Learning. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
39. Das, R.; Dukler, Y.; Ravichandran, A.; Swaminathan, A. Learning Expressive Prompting With Residuals for Vision Transformers. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
40. Li, Z.; Peng, B.; He, P.; Galley, M.; Gao, J.; Yan, X. Guiding Large Language Models via Directional Stimulus Prompting. *arXiv* **2023**, arXiv:2302.11520.
41. Zheng, C.; Liu, Z.; Xie, E.; Li, Z.; Li, Y. Progressive-Hint Prompting Improves Reasoning in Large Language Models. *arXiv* **2023**, arXiv:2304.09797.
42. Bulat, A.; Tzimiropoulos, G. LASP: Text-to-Text Optimization for Language-Aware Soft Prompting of Vision & Language Models. In Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Vancouver, BC, Canada, 17–24 June 2023.
43. Rasool, A.; Shahzad, M.I.; Aslam, H.; Chan, V.; Arshad, M.A. Emotion-Aware Embedding Fusion in Large Language Models (Flan-T5, Llama 2, DeepSeek-R1, and ChatGPT 4) for Intelligent Response Generation. *AI* **2025**, *6*, 56. [CrossRef]
44. Levi, E.; Brosh, E.; Friedmann, M. Intent-based Prompt Calibration: Enhancing prompt optimization with synthetic boundary cases. *arXiv* **2024**, arXiv:2402.03099.
45. Huang, Z.; Chen, K.; He, J.; Bai, X.; Karatzas, D.; Lu, S.; Jawahar, C.V. ICDAR2019 Competition on Scanned Receipt OCR and Information Extraction. In Proceedings of the 2019 International Conference on Document Analysis and Recognition (ICDAR), Sydney, Australia, 20–25 September 2019.
46. Palshikar, G.K.; Pawar, S.S.; Banerjee, A.S.; Srivastava, R.; Ramrakhiyani, N.; Patil, S.; Thosar, D.; Bhat, J.M.; Jain, A.; Hingmire, S.; et al. RINX: A system for information and knowledge extraction from resumes. *Data Knowl. Eng.* **2024**, *147*, 102202. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.