Augmenting Intelligent Document Processing (IDP) Workflows with Contemporary Large Language Models (LLMs)

Article *in* International Journal of Computer Trends and Technology · October 2023

DOI: 10.14445/22312803/IJCTT-V71I10P110

1 author:

Shreekant Mandvikar
Ally Bank
8 PUBLICATIONS **56** CITATIONS

SEE PROFILE

International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V71I10P110

Volume 71 Issue 10, 80-91, October 2023

Original Article

Augmenting Intelligent Document Processing (IDP) Workflows with Contemporary Large Language Models (LLMs)

Shreekant Mandvikar

Principal Automation Engineer, Ally Bank, Charlotte, NC, USA.

[1]*Corresponding Author : shreekant.mandvikar@gmail.com*

Received: 27 August 2023 Revised: 29 September 2023 Accepted: 11 October 2023 Published: 30 October 2023

Abstract *- The current decade has witnessed an explosion in the volume of documents generated by businesses, academic institutions, and other organizations. Managing, analyzing, and extracting value from this vast array of documents has become challenging. Integration of Large Language Models (LLMs) into intelligent document processing can significantly address this challenge. This research explores the contributions of Large Language Models (LLMs) in enhancing the various stages of the Intelligent Document Processing (IDP) workflow. Specifically, how LLMs can enhance each step of the IDP offered on AWS. In the initial document classification stage of the workflow, LLMs can offer an improved semantic-based and hierarchical classification of documents. However, this can introduce challenges such as overfitting, bias, and increased computational overhead. During the document extraction stage, LLMs provide benefits in contextual interpretation, cross-referencing data, and data transformation. In the review & validation stage, LLMs can augment human efforts by offering automated suggestions and anomaly detection, although this can sometimes result in false alarms. In the document enrichment stage, LLMs contribute by offering contextual enrichment, better sentiment analysis, and topic modeling but risk over-enriching data. In the data integration stage, LLMs can synthesize data for consistency, generate automated narratives, and facilitate API interactions for smoother integration. Across these different stages, LLMs are subject to limitations like increased computational costs, dependency on training data for specialized tasks, and latency in real-time operations.*

Keywords *- Artificial intelligence-driven document enrichment, Intelligent document classification, Intelligent data extraction, Intelligent document processing, Large language models, Semantic understanding.*

1. Introduction

Organizations worldwide interact with their customers, and others share data in multiple ways. Unlike traditional interfaces, direct data entry documents are the largest source of incoming data to organizations, and their applications are documents, for example, purchase orders, driving licenses, invoices, etc. Intelligent Document Processing (IDP) has revolutionized how applications interact with vast troves of information. By leveraging the capabilities of large language models, IDP transcends traditional boundaries, automating data extraction and understanding intricate content nuances. As digital transformation sweeps across sectors, the need for extracting valuable insights from unstructured documents grows exponentially. With their deep learning foundations, large language models enable machines to comprehend text similarly to humans but at unparalleled speeds. This convergence of intelligence and automation offers unprecedented opportunities for businesses, researchers, and individuals, streamlining operations and paving the way for innovation in document-centric tasks.

2. Large language Models

Large language models are predominantly built on a class of deep learning architectures called transformer networks. Initially proposed by Vaswani et al. in 2017 [1], transformers have become fundamental for natural language processing tasks. These architectures are particularly adept at understanding the context and semantics of sequential data, including but not limited to text. A typical transformer network consists of multiple transformer blocks or layers, each contributing to the overall functioning of the model. These layers generally include self-attention mechanisms, feed-forward neural networks, and normalization layers. Self-attention allows the model to weigh the importance of different parts of the input, while feed-forward layers and normalization layers help in computation and stabilization, respectively. By stacking these layers, one can build increasingly deep and powerful transformer models, which are subsequently capable of more complex tasks [2], [3]. Stacking layers enhances the model's capacity to learn from data, enabling better predictions and interpretations during inference.

2.1. How Large Language Model Works

Large Language Models (LLMs) are predicated on statistical methodologies that encode a probabilistic relationship among sequences of words [4], [5]. These models use a probability distribution denoted as P(w1,...,wL), which aims to approximate the empirical distribution observed in a substantial corpus of text in a specific language. The simplest form of such a model is the "1-gram" model, as given below, which operates under the assumption that words are independently distributed [6], [7].

$$P(w_1, \dots, w_L) = \prod_{i=1}^{L} P(w_i); \quad P(W) = \frac{\mathrm{n}(w)}{W}$$

Where n(w) and W represent the number of occurrences of w in the corpus and the total number of words in the corpus.

Where the probability of a word sequence P(w1,..., wL) is computed as the product of the probabilities of individual words P(wi), the individual word probabilities are determined by the frequency of each expression in the corpus relative to the total number of words in that corpus [8], [9]. To assess the effectiveness of a language model, the standard metric employed is cross entropy, which quantifies how well the model's probability distribution mirrors the empirical distribution observed in the corpus [10], [11]—the cross-entropy.

L is calculated as a sum of logarithmic terms related to the conditional probabilities of word sequences, often denoted as:

$$L = -\frac{1}{N}\sum_{i=1}^{N-n} \log P\left(W_{i+n} \mid W_i, W_{i+1}, \dots, W_{i+n-1}\right)$$

Where the perplexity is represented as exp(−L). The equation is an objective function to minimise during the training phase. Various optimization techniques, such as backpropagation, can be employed.

2.2. Comparison of Models

Below table 1 highlights a comparison of traditional Machine Learning (ML), Deep Learning (DL), and Large Language Models (LLMs) [12]. In terms of data requirements, traditional ML models generally require data in the thousands to the millions range. At the same time, DL and LLMs demand significantly more, extending to billions of data points. Traditional ML often requires manual, domain-specific intervention for feature engineering, whereas DL and LLMs can automatically extract meaningful features.

Table 1. Comparison among traditional Machine Learning (ML), Deep Learning (DL), and Large Language Models (LLMs)

Comparison	Traditional ML	Deep Learning	Large Language Models
Training Data Size	Large (Thousands to Millions)	Large (Thousands to Millions)	Very Large (Billions+)
Feature Engineering	Manual (Domain expertise features)	Automatic (Self-learns features)	Automatic (Self-learns required)
Model Complexity	Limited (Linear, Tree-based)	Complex (Convolutional, Recurrent)	Very Complex (Up to billions of parameters)
Interpretability	Good (Transparent algorithms)	Poor (Black-box nature)	Poorer (Extremely complex, black-box nature
Performance	Moderate (Sufficient for simpler tasks)	High (Effective for complex for multiple tasks)	Highest (State-of-the-art tasks)
Hardware Requirements	Low (CPUs sufficient)	High (GPUs often required)	Very High (Multiple GPUs, TPUs often required)
Computational Cost	Low to Moderate	High	Very High
Real-Time Capabilities	Often Suitable	Less Suitable (due to Complexity)	Generally Not Suitable
Adaptability	Lower (Fine-tuning often leaning)	Moderate (Some transfer learning)	High (Very adaptable with needed)
Software Libraries	Scikit-learn, Stats models	TensorFlow, PyTorch	Hugging Face Transformers, GPT-specific implementations

While traditional ML models are often limited in complexity, employing linear or tree-based algorithms, DL and LLMs are far more intricate, incorporating complex architectures like convolutional and recurrent neural networks and billions of parameters in the case of LLMs. Interpretability remains a strong point for traditional ML, with many of its algorithms being transparent enough for straightforward interpretation, in contrast to the black-box

nature of DL and LLMs. Traditional ML offers moderate performance capabilities, usually sufficient for shorter tasks. In contrast, DL is well-suited for more complex tasks, and LLMs frequently achieve state-of-the-art results across multiple domains. Hardware requirements scale accordingly, from the low conditions of traditional ML, which can often run efficiently on CPUs, to the high and very high demands of DL and LLMs, which usually necessitate specialized hardware like GPUs and TPUs. Additional aspects such as computational cost, real-time capabilities, and adaptability differ significantly. Traditional ML generally has lower computational costs and is often suitable for real-time applications, while DL and LLMs are more computationally intensive and less suited for real-time tasks. However, LLMs and DL models are more adaptable and often capable of transferring learning across different studies, making them more flexible for a broader range of applications.

2.3. Classes of LLM

Large Language Models (LLMs) can be categorized into several classes, each with unique capabilities for different use cases, as shown in below table 2. Autoregressive models, such as GPT-3 (Generative Pretrained Transformer 3), are compelling at generating human-like text, making them ideal for tasks such as content creation, conversational agents, and creative writing [13], [14]. Another class consists of Encoder-only models like BERT (Bidirectional Encoder Representations from Transformers), designed to understand and represent textual data rather than generate it. These are commonly used in text classification, sentiment analysis, and information retrieval tasks. A third class comprises Encoder-Decoder models like T5 (Text-to-Text Transformer) or BART (Bidirectional and

Auto-Regressive Transformers), which are versatile in both understanding and generating text. These models are particularly useful for machine translation, summarization, and question-answering systems. Moreover, specialized models like ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) are designed for efficiency and are suitable for tasks that demand lower computational resources but require high performance. The choice among these classes depends on the specific requirements of the task at hand, including but not limited to performance expectations, computational resources, and the nature of the data to be processed.

In the contemporary digital age, the accumulation of documents is undergoing exponential growth, necessitating rapid and accurate processing methods to extract meaningful value. Traditional document processing techniques are increasingly inadequate for addressing this escalating need for speed and precision. Legacy systems, built on older technologies, were not designed to manage the vast amounts of data that modern businesses encounter. These archaic methods frequently employ batch processing, which lacks the capability for real-time analytics and fails to accommodate the increasing velocity of incoming data streams. This inefficiency contributes to data bottlenecks.

Table 2. Overview of different classes of large language models

Model Class	Short Description	Typical Use Cases	Example Models
Autoregressive	Specialized in generating coherent and contextually relevant text.	Content creation, Conversational agents	GPT-3 (Generative Pretrained Transformer 3)
Encoder-only	Optimized for understanding and representing text rather than generating it.	Text classification, Sentiment analysis	BERT (Bidirectional Encoder Representations from Transformers)
Encoder-Decoder	Capable of both understanding and generating text, offering versatility.	Machine translation, Summarization	T5 (Text-to-Text Transformer), BART (Bidirectional and Autoregressive Transformers)
Specialized	Designed for specific tasks that require high performance but lower computational costs.	Efficient text classification, Information retrieval	ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately)

Large Language Models (LLMs) can be categorized into several classes, each with unique capabilities for different use cases, as shown in Table 2. Autoregressive models, such as GPT-3 (Generative Pretrained Transformer 3), are compelling at generating human-like text, making them ideal for tasks such as content creation, conversational agents, and creative writing [13], [14]. Another class consists of Encoder-only models like BERT (Bidirectional Encoder Representations from Transformers), designed to understand and represent textual data rather than generate it. These are commonly used in text classification, sentiment analysis, and information retrieval tasks. A third class comprises Encoder-Decoder models like T5 (Text-to-Text Transformer) or BART (Bidirectional and Auto-Regressive Transformers), which are versatile in both understanding and generating text. These models are particularly useful for machine

translation, summarization, and question-answering systems. Moreover, specialized models like ELECTRA (Efficiently Learning an Encoder that Classifies Token Replacements Accurately) are designed for efficiency and are suitable for tasks that demand lower computational resources but require high performance. The choice among these classes depends on the specific requirements of the task at hand, including but not limited to performance expectations, computational resources, and the nature of the data to be processed. In the contemporary digital age, the accumulation of documents is undergoing exponential growth, necessitating rapid and accurate processing methods to extract meaningful value. Traditional document processing techniques are increasingly inadequate for addressing this escalating need for speed and precision. Legacy systems, built on older technologies, were not designed to manage the vast amounts of data that modern businesses encounter. These archaic methods frequently employ batch processing, which lacks the capability for real-time analytics and fails to accommodate the increasing velocity of incoming data streams. This inefficiency contributes to data bottlenecks and delayed decision-making, affecting the organisation's overall performance [15].

Furthermore, manual document handling and paper-based processes continue to be a stumbling block, impeding the efficiency of internal operations and customer-facing services. Manual data extraction is a labor-intensive activity susceptible to human error, leading to inaccuracies and potential non-compliance with industry standards or regulations. These inefficiencies are magnified by data in structured forms, such as databases and spreadsheets, and unstructured forms, like emails, social media posts, and other types of textual content. Manual processing can neither scale to handle this diversity nor ensure the high level of data security that automated solutions can offer [16].

The administrative overhead in manual data extraction and processing also considerably drains employee productivity. Given that significant time and resources are spent on routine tasks such as data entry, ticket routing, and workflow management, less time is available for employees to focus on strategic, value-added activities. This is particularly concerning in environments where rapid data processing is vital for competitive advantage or regulatory compliance. As the volume and complexity of data continue to rise, the limitations of manual and legacy processing methods will only become more pronounced, necessitating a shift towards automated, intelligent document processing solutions. Intelligent Document Processing (IDP) uses Artificial Intelligence (AI) technologies for the automatic extraction and processing of data from an array of document types [17], [18].

Unlike traditional systems that often require predefined templates for data extraction, IDP operates in a "template-free mode," offering the flexibility to handle both structured and unstructured data. The technology stack behind IDP usually incorporates Optical Character Recognition (OCR) for converting different types of written or printed characters into machine-encoded text. Natural Language Processing (NLP) is also employed to understand the context and semantics of the text, enabling more accurate data extraction. Machine Learning (ML) algorithms are also trained to continuously improve the system's performance, adapting to new data structures and layouts. These AI-driven capabilities significantly enhance the efficiency and accuracy of data extraction. Alongside the core data extraction functionalities, IDP also incorporates a variety of pre-processing and post-processing operations. Pre-processing might include functions like document classification, segmentation, and noise reduction to prepare the data for more effective extraction. On the other hand, post-processing involves tasks such as data validation, transformation, and normalization to ensure that the extracted data is ready for downstream applications or analytics. These additional steps contribute to enhancing the quality and reliability of the extracted.

3. LLMs in IDP workflow

Document capture is an initial step in the Intelligent Document Processing (IDP) workflow, and Amazon Web Services (AWS) offers robust solutions for this phase. One of the primary AWS services utilized for this purpose is Amazon Simple Storage Service (S3). Users can upload documents in multiple formats like PDF, JPEG, PNG, and TIFF to an S3 bucket. Amazon S3 is a highly scalable, reliable, and low-latency data storage service, allowing organizations to store vast data securely. The architecture of S3 is designed to accommodate the high-speed ingestion of documents from various sources, be it automated data pipelines, manual uploads, or integrations with other platforms. This flexibility ensures that organizations can efficiently manage the intake of documents without worrying about infrastructure limitations or bottlenecks [19]. The role of the S3 bucket in this workflow extends beyond mere storage; it acts as a centralized repository for documents, facilitating easier management and subsequent processing. Single and multi-page documents of varying types can be stored cohesively, allowing for streamlined categorization and retrieval. Advanced features like versioning and lifecycle policies can be implemented for better document management.

3.1. Document Classification Stage

In the Intelligent Document Processing (IDP) workflow, the Document Classification stage is used for automating the sorting and processing of different types of documents. Amazon Web Services (AWS) provides Amazon Textract and Amazon Comprehend. Amazon Textract helps extract text and other data from scanned documents. Its machine learning models can recognise various formats and layouts, thus enabling the accurate capture of data points within the papers. On the other hand, Amazon Comprehend employs

natural language processing (NLP) techniques to understand the context and semantic details of the textual data extracted.

Both services offer capabilities for training machine learning models, allowing organizations to tailor the classification to meet specific needs. These training features make it possible to create highly accurate classification models for documents such as contracts, invoices, and receipts. Once the training phase is complete, the inference stage comes into play. Amazon Textract and Amazon Comprehend work in tandem to classify incoming documents automatically. Amazon Textract extracts the necessary text and data, fed into Amazon Comprehend for contextual analysis and categorization.

Large Language Models (LLMs) can significantly enhance Document Classification in several ways. One of the primary advantages is their capability for semantic understanding. Unlike traditional keyword-based methods that often rely on surface-level text matching, LLMs can analyze the content of documents to understand the underlying intent and meaning. This allows for a more robust and accurate classification, especially for records where the intended category is not immediately apparent from the terminology used. For instance, a legal contract may not always contain the expected keywords. However, an LLM can still classify it correctly based on understanding the document's overall content and structure.

In complex document ecosystems, it is often necessary to categorize documents into multiple layers of taxonomy, which could be hierarchical or nested in nature. LLMs can facilitate this by analyzing a document, determining its primary category, and where it fits within subcategories. This level of granularity is beneficial in sectors like healthcare, law, and finance, where documents often need to be sorted into intricate categorization systems for better retrieval and compliance.

One issue is the potential for overfitting and bias. Suppose the training data used to develop the LLM are not sufficiently diverse or are skewed towards specific categories. In that case, the model may perform poorly when exposed to different types of documents. It may also inherit biases in the training data, leading to problematic classifications. Additionally, the computational overhead for LLMs can be significant, especially when compared to more straightforward keyword-based classification methods. Semantic understanding and hierarchical classification generally require more computational power and time, which could be a limiting factor for organizations without robust computing resources.

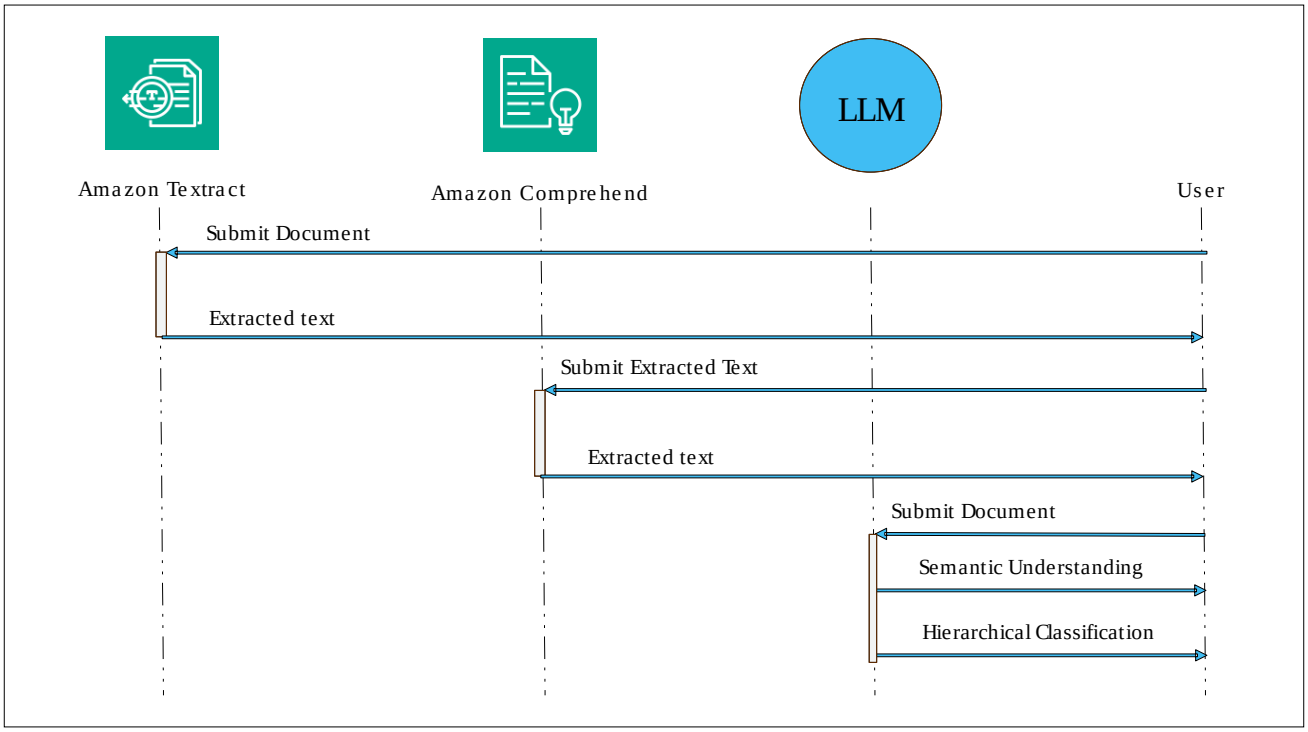

Fig. 1 Contributions of LLMs in the classification stage of IDP

As shown in Figure 1, the workflow starts when a user submits a document to Amazon Textract for text extraction. Upon receiving the copy, Amazon Textract becomes operational and carries out its designated text extraction task from the paper. Once the text has been successfully extracted, Amazon Textract sends this information back to the user and subsequently deactivates it, signaling the conclusion of its role in the process.

Subsequently, the user submits the extracted text to Amazon Comprehend for primary document classification. Upon activation, Amazon Comprehend categorizes the document into predetermined types, such as contracts, invoices, or receipts. This preliminary classification result is then transmitted back to the user, and Amazon Comprehend is deactivated, thereby ending its participation in the workflow.

In parallel, the user can utilize Large Language Models (LLMs) for more advanced classification techniques. LLMs offer users two distinct and more nuanced document classification methods when activated. The first method employs semantic understanding, wherein the LLM scrutinizes the document's content to understand its meaning and intent, going beyond mere keyword matching. The second method involves hierarchical classification. In this approach, the LLM categorizes the document at a basic level and organizes it into multi-level or hierarchical categories based on its content.

Upon completing these advanced classification tasks, the LLMs send the enriched classification results back to the user. The LLMs are then deactivated, concluding their role in the document classification.

3.2. Document Extraction Stage

The Document Extraction stage enables organizations to gather valuable information from the classified documents for further processing or analysis. Amazon Textract also plays a significant role in this phase by offering Application Programming Interfaces (APIs) that facilitate the extraction of structured and unstructured data [20]. The service can handle various document types, including invoices, receipts, and identity documents. Textract's API functions can target specific queries within documents, allowing users to zero in on particular data fields such as dates, invoice numbers, or line items. This level of detail is essential for many business applications, such as accounts payable automation or customer data management, where precision and accuracy are paramount.

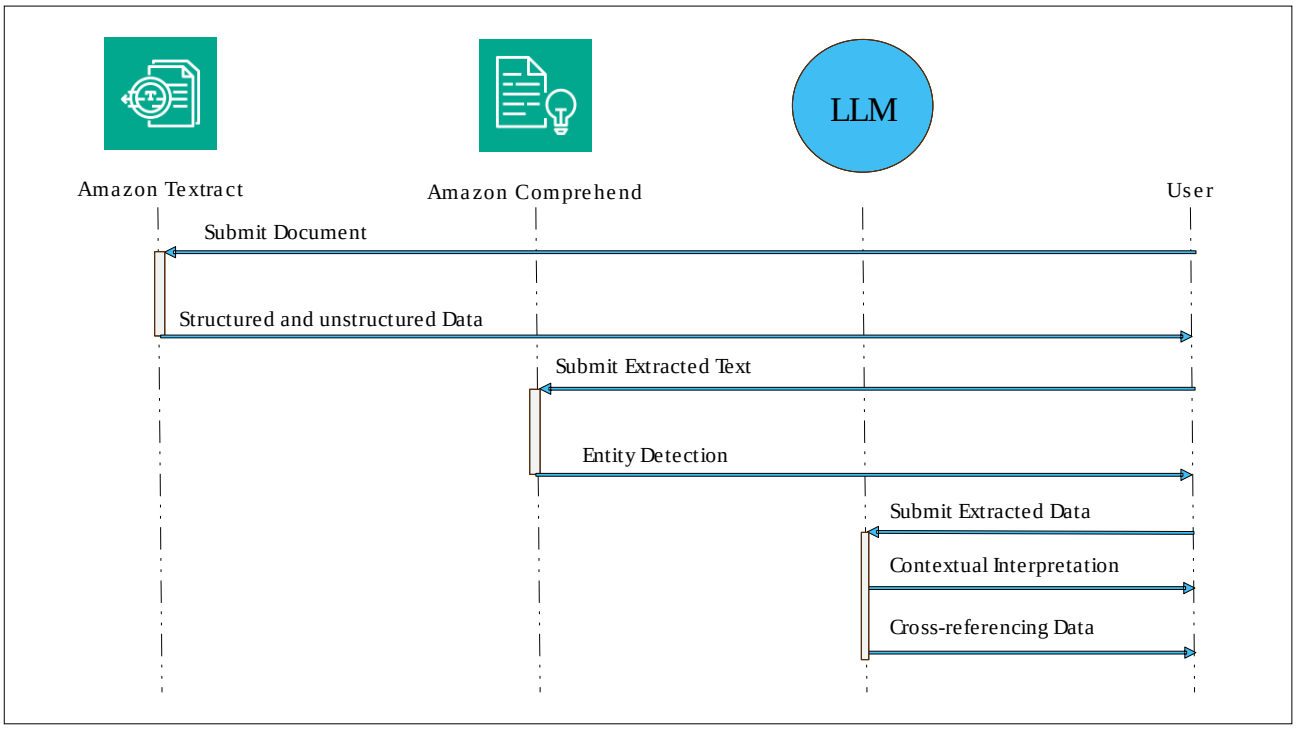

Fig. 2 Contributions of Large Language Models (LLMs) in the extraction stage of IDP

To enhance the capabilities of Amazon Textract, Amazon Comprehend offers complementary features through its own set of APIs. Specifically, Amazon Comprehend excels in entity detection, which involves identifying specific items like names, dates, or organizations within the text. Additionally, it provides tools for training and deploying custom entity recognizers, enabling organizations to identify industry-specific terms or codes that may not be part of a general language model. By using Amazon Comprehend in conjunction with Amazon Textract, users can extract raw text and understand the context and significance of the extracted data. The synergy between these two services provides a robust and versatile solution for the Document Extraction stage, ensuring that the extracted data is comprehensive and contextually relevant.

Traditional extraction methods may identify and pull data but cannot often understand the context in which that data exists within the document. On the other hand, LLMs can infer the underlying meaning or importance of extracted

data by analyzing its context. This becomes particularly crucial when dealing with ambiguous or vague data points. For example, suppose a financial document mentions a number. In that case, an LLM can help determine whether that number represents revenue, a liability, or something else entirely based on the surrounding text and structure of the document.

Traditional extraction methods may treat each piece of data as an isolated entity, but LLMs can establish links between different parts of a document. This is useful for creating a cohesive understanding of the document's content. For instance, if a contract mentions terms in multiple sections, an LLM can cross-reference these terms to ensure they are consistently interpreted, thereby increasing the accuracy and reliability of the extracted information.

Once data is extracted, it must often be converted into different formats or narratives to be applicable across diverse applications. LLMs can automatically transform extracted data into various forms, such as tables JSON objects, or even generate summaries, facilitating easier integration into databases or other systems.

This feature is handy in sectors like healthcare, where extracted data may need to be presented in standardized formats for electronic health records. Overall, LLMs offer a more intelligent and adaptable approach to document extraction, although it is essential to consider factors like computational cost and the quality of the training data.

The workflow starts with the user submitting a document to Amazon Textract. Once activated, Amazon Textract extracts structured and unstructured data from the document. This could include various types of information, such as text from specific queries, invoices, or receipts. After the extraction, Amazon Textract returns and deactivates the extracted data to the user. The user then sends the extracted data to Amazon Comprehend. Upon activation, Amazon Comprehend detects entities on the data, identifying specific items or concepts within the text. After finishing this task, it returns the entity detection results to the user and deactivates. Optionally, the user can further enhance the extracted data by submitting it to Language Learning Models (LLMs).

A note emphasises LLMs' specialised roles in improving the document extraction process. Once activated, LLMs offer two main enhancements: a)Contextual Interpretation: LLMs can provide context to ambiguous or vague data extracted. This deepens the understanding and utility of the extracted data. b)Cross-referencing Data: LLMs can also correlate or link data extracted from different document parts, providing a more cohesive understanding of the information. After providing these enhanced extraction capabilities, LLMs deactivate, signaling the end of their role in the process.

3.3. Review & Validation Stage

While machine learning models like Amazon Textract and Amazon Comprehend are efficient in extracting and classifying data, they may not consistently achieve 100% accuracy due to document complexities and variations [21]. A2I facilitates a human review workflow, where a human workforce can validate and correct the extracted information. The service integrates seamlessly with other AWS services, enabling organizations to easily include a manual review step in their automated document processing pipelines. It provides features for task assignment, data annotation, and review outcomes, all designed to ensure that the extracted information meets the desired level of accuracy and completeness.

In addition to human validation, AWS Lambda is also used in post-processing activities. Lambda is a serverless computing service that allows users to run code without provisioning or managing servers. In the context of IDP, Lambda functions can be triggered automatically once the data extraction and human review phases are completed. These functions can perform various tasks, such as post-processing checks and rule-based validations, to ensure that the extracted data conforms to predefined standards or business rules. For instance, Lambda can be configured to validate invoice numbers against a database or check that mandatory contract fields are filled in.

Within Amazon's Augmented AI (A2I) workflow, Large Language Models (LLMs) can introduce several features in the Review and validation stages. The first is the provision of automated suggestions. When human reviewers assess documents, real-time suggestions or corrections generated by an LLM can significantly aid the process. For instance, if a reviewer looks at medical records, an LLM could modify medical terms or drug names that appear inconsistent or misspelt. By doing so, LLMs can help minimize human errors, increase the speed of the review process, and enhance overall accuracy.

Rather than requiring human reviewers to sift through each data point manually, LLMs can flag potential anomalies automatically. This can be particularly useful in complex documents where human oversight could easily miss inconsistencies or errors. By flagging these potential issues, LLMs help human reviewers focus on problematic areas, making the review process more efficient and targeted. For example, an LLM could flag transactions deviate significantly from established patterns in a financial audit scenario, allowing auditors to focus on those particular entries.

One issue can be the possibility of false alarms in anomaly detection. While LLMs can identify potential anomalies, they may sometimes flag data points that are not problematic, thus increasing the workload for human

reviewers. This can add unnecessary steps to the review process and may require additional resolution time. Sometimes, it could also lead to human reviewers becoming desensitized to the flags, potentially ignoring genuine anomalies.

AWS Lambda starts by sending extracted data to Amazon A2I for review. At the review stage, Amazon A2I can either directly present the data to a human reviewer or use LLMs for automated checks. During the human review workflow, the reviewer may request real-time suggestions from the LLMs. The LLMs then provide these suggestions to assist the human reviewer, who validates and submits the review. Alternatively, Amazon A2I may request anomaly detection from the LLMs. LLMs flag potential anomalies and return this information to A2I. The flagged data is then presented to the human reviewer for validation. Once the review process is complete, Amazon A2I sends the validated data back to AWS Lambda for post-processing checks and rule-based verifications.

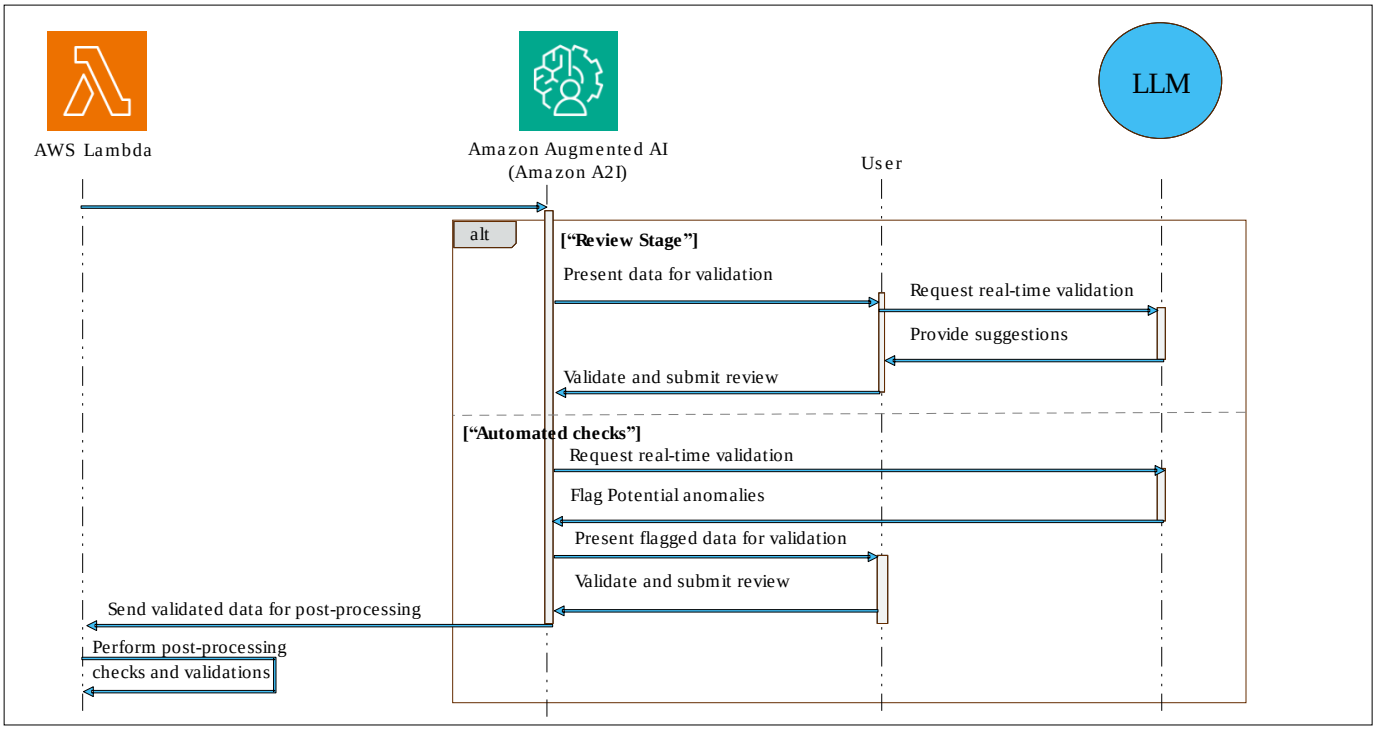

Fig. 3 Contributions of Large Language Models (LLMs) in the review/validation stage of IDP

3.4. Document Enrichment Stage

The Intelligent Document Processing (IDP) workflow is an enrichment phase that focuses on adding value to the extracted data by employing advanced analytics and specific actions. Amazon Comprehend is often utilized for enriching general data, leveraging its natural language processing capabilities to analyze and categorize text. Amazon Comprehend Medical handles sensitive and complex medical data for specialised sectors like healthcare. These services can analyze large sets of documents quickly and accurately, providing deep insights into the content [22], [23]. Whether identifying sentiments in customer feedback forms or extracting medical entities from clinical notes, the enrichment phase enables organizations to derive more meaningful information from their data corpus.

One of the critical activities in the enrichment phase is the implementation of various actions to secure, tag, and manage the documents. Measures may include redacting personally identifiable information (PII) or protected health information (PHI) to comply with regulations such as the General Data Protection Regulation (GDPR) or the Health Insurance Portability and Accountability Act (HIPAA). Additionally, the phase often involves tagging documents for easier categorization and retrieval, adding metadata for enhanced searchability, and implementing legal holds for documents subject to legal proceedings or investigations. These enrichment actions augment the extracted data's utility and contribute to fulfilling compliance and governance requirements. Organizations can effectively manage various data types and complexities by integrating Amazon Comprehend and Amazon Comprehend Medical into this phase.

In the Document Enrichment phase of intelligent document processing, Large Language Models (LLMs) can provide enhancements beyond essential data extraction and validation. One such feature is contextual enrichment. LLMs can offer context and additional insights for the extracted data by leveraging their extensive knowledge base. For

example, if an LLM extracts a set of dates and names from a historical document, it could also provide contextual information, such as the significance of those dates or the roles of the individuals mentioned, making the data more valuable and informative.

For documents where understanding the emotional tone is essential, such as customer reviews or employee surveys, LLMs can automatically gauge the sentiment expressed in the text. This offers an extra layer of information crucial for businesses aiming to improve customer satisfaction or employee engagement. The sentiment scores or labels can be included as additional metadata, meaningfully enriching the document.

Additionally, LLMs can engage in topic modeling to identify overarching themes or subjects within the documents. This form of enrichment helps in categorizing documents more effectively and can be particularly useful for large document corpora where quick information retrieval is essential. For example, in a collection of research papers, an LLM could identify main themes like "Artificial Intelligence," "Climate Change," or "Public Health," which could then be used as metadata for more straightforward navigation and search. However, one limitation to consider is the risk of over-enrichment. While LLMs can provide extensive contextual data and insights, there is a potential for information overload. If too much additional information is appended, it can complicate the document and make it challenging for end-users to understand the primary content quickly.

The process can start with a general data enrichment process using Amazon Comprehend. Decision points are used to determine if additional enrichment steps, such as contextual enrichment, topic modeling, or sentiment analysis, are needed. If these are required, Amazon Comprehend requests these services from LLMs and receives the processed information. Finally, Amazon Comprehend adds metadata and implements legal holds on specific documents, completing the Document Enrichment stage.

3.5. Data Integration Stage

After the data has been extracted, validated, and enriched, the next step in the Intelligent Document Processing (IDP) workflow involves storage and integration. Amazon Simple Storage Service (S3) is commonly used as a destination for storing the finalized data [24], [25]. The advantage of using Amazon S3 for this purpose is its scalability, security features, and compatibility with various data types and formats. Organizations can organize this data within the S3 bucket to align with their operational or analytical needs.

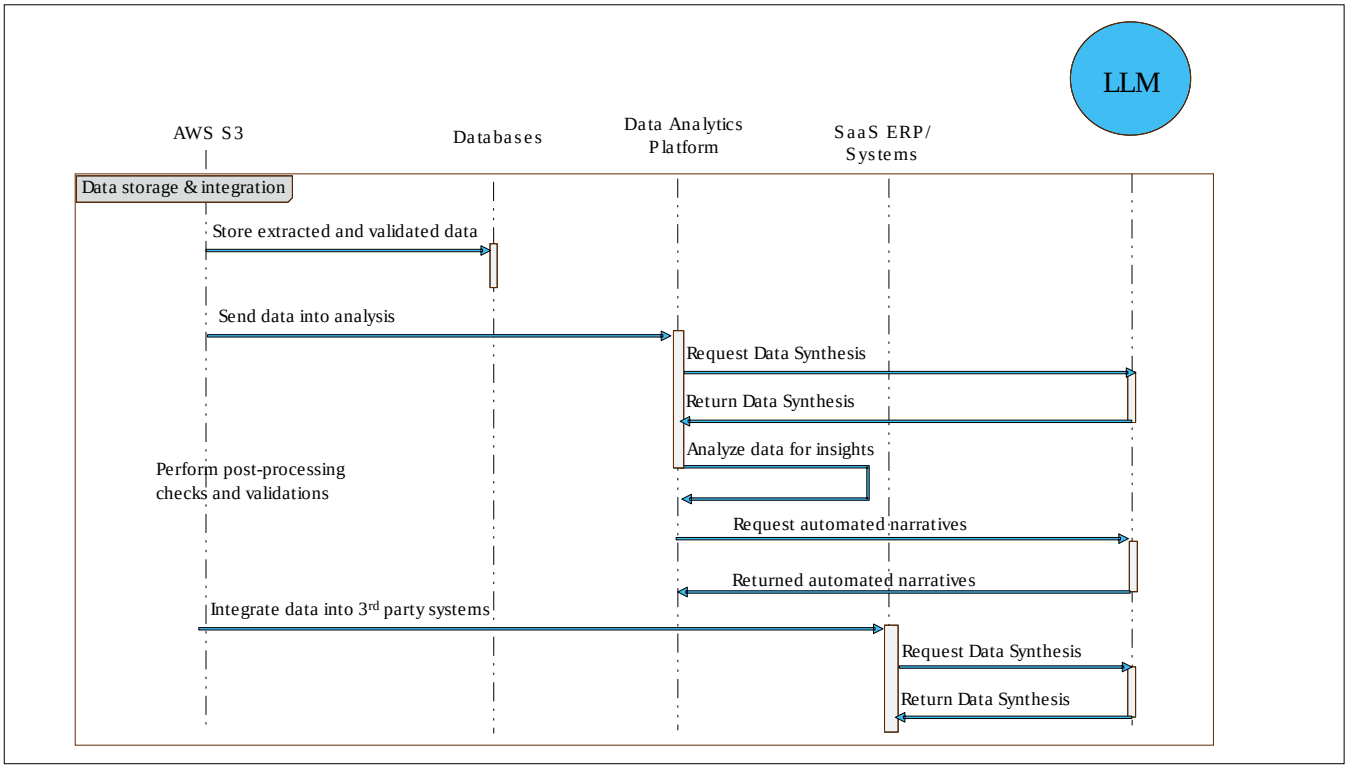

Fig. 4 Contributions of Large Language Models (LLMs) in the review/validation stage of IDP

Alternatively, the data can be integrated into databases, whether they are relational databases like Amazon RDS, NoSQL databases like Amazon DynamoDB, or data warehousing solutions like Amazon Redshift. This allows for easier querying and data manipulation capabilities, providing a solid foundation for downstream applications requiring this data.

The utility of the extracted and validated data extends beyond mere storage; it serves as valuable input for analytics and third-party integrations. Data Analytics platforms can pull this data to perform myriad analyses to derive insights. These platforms can range from Amazon Quick Sight for business intelligence to more specialized analytics software designed for specific industries or use cases.

Furthermore, the data can be integrated with Software as a Service (SaaS) systems or third-party applications for various purposes. Whether it is an Enterprise Resource Planning (ERP) system that needs invoice data for financial accounting or web/mobile applications that utilize customer information for personalized experiences, the integration possibilities are extensive. The IDP workflow, therefore, not only aids in the initial stages of capturing and processing data but also enables organizations to maximize the utility of this data in numerous applications and analyses.

One key feature is data synthesis. Data extracted from multiple documents or sources must often be integrated into a single database or analytics platform. LLMs can facilitate this process by synthesizing the data to ensure consistency and coherence. For instance, if similar data points are extracted from different types of contracts, an LLM can harmonize those data points to create a standardized dataset suitable for integration. This results in more reliable and actionable insights when the integrated data is later analyzed.

Another potential enhancement is the generation of automated narratives. Particularly useful for analytics platforms, LLMs can provide summaries or narratives based on the integrated and analyzed data, making it easier for decision-makers to comprehend the insights drawn from the data. For example, suppose an analytics platform is used to assess sales performance. In that case, an LLM can automatically generate a summary that describes crucial trends, outliers, or noteworthy events using the integrated data. These narratives can complement visual analytics, such as charts or graphs, and can be especially useful for stakeholders who may not be experts in data analysis.

The sequence starts with storing extracted and validated data in Amazon S3 and databases. Next, the data is sent to Data Analytics platforms for analysis. Before conducting the research, LLMs synthesise the data, ensuring consistency and coherence.

Data Analytics platforms also request automated narratives from LLMs after analyzing the data for insights. This offers a natural language summary or interpretation of the analytics, providing an additional enrichment layer. The data is then integrated into SaaS or ERP systems. Like the analytics platforms, these systems also engage LLMs to synthesize the data before it's used or processed.

4. Conclusion

Businesses today face significant challenges as they are under constant market pressures to increase efficiency, enhance the customer experience, reduce operational costs, and ensure compliance with regulatory standards. One of the contributing factors to these pressures is the overwhelming volume of data entering enterprise systems daily. This data, which includes semi-structured documents like contracts and invoices and unstructured ones like handwritten letters and emails, is rapidly growing daily.

The technology behind IDP includes Optical Character Recognition (OCR) for scanning documents, Natural Language Processing (NLP) for understanding language-based data, computer vision for identifying patterns, and Machine Learning (ML) and Artificial Intelligence (AI) for continuous learning and optimization of the processing system [26]–[28]. With these technologies combined, IDP solutions can recognize and understand a wide range of document formats, extracting relevant data and integrating it into business processes and downstream systems.

This showed how Large language models (LLMs) can enhance the various stages of the intelligent document processing (IDP) workflow. Starting with the document classification stage, where Amazon Textract and Amazon Comprehend are commonly used, LLMs can bring semantic understanding and hierarchical classification into play, going beyond traditional keyword matching. However, they may face challenges such as bias and computational demands.

LLMs excel in contextual interpretation in the document extraction stage, improving data extraction accuracy by deciphering ambiguous information. They also facilitate cross-referencing data from different document sections and aid in data transformation for versatile applications. In the review and validation stage, LLMs streamline the process by offering real-time suggestions and anomaly detection, though there is a need to mitigate false alarms. In the document enrichment stage, LLMs provide contextual enrichment, sentiment analysis, and topic modeling, adding value but requiring vigilance against over-enrichment. Finally, in the data integration stage, LLMs ensure data consistency and generate automated narratives, enhancing the overall efficiency of the IDP workflow.

In utilising Large Language Models (LLMs) across various stages of Intelligent Document Processing (IDP), the

first consideration to address is the computational costs. Applying LLMs, especially those designed for complex semantic understanding or anomaly detection tasks, necessitates substantial computational resources. These resources are essential for the model's operation, including but not limited to forward and backward passes during the inference and learning phases. The cost implication of these computational requirements can be significant, particularly for organizations without robust computational infrastructure. Consequently, a thorough cost-benefit analysis becomes imperative to assess whether implementing LLMs aligns with budgetary constraints and long-term financial strategy.

While pre-trained models are endowed with extensive general knowledge, domain-specific tasks or idiosyncratic nuances may necessitate the application of specialized training data to fine-tune these models. The quality and quantity of such training data become critical variables in the performance efficacy of the model. For example, in a legal context, an LLM would require exposure to various legal documents, terminologies, and precedents to accurately classify or extract information from legal contracts or judicial opinions. Thus, generating or acquiring domain-specific training data is an essential consideration regarding logistics and costs.

The processing time involved in running large neural network models can introduce latency into the workflow, adversely affecting time-sensitive operations. For instance, in the context of real-time analytics or fraud detection, the introduction of even slight delays can have significant ramifications. Such delays may be accentuated if the model needs to synthesize data from multiple sources or generate automated narratives.

Moreover, addressing the limitations posed by potential false alarms in anomaly detection or information overload in data enrichment adds another layer of complexity. Fine-tuning the model to minimize false positives without compromising on detecting genuine anomalies calls for meticulous algorithmic adjustments and validation protocols. Similarly, designing the LLM to provide just the right amount of contextual enrichment without overwhelming the user demands an intricate balance of natural language processing capabilities and user experience design.

References

[1] Ashish Vaswani et al., "Attention is All You Need," *Advances in Neural Information Processing Systems*, 2017. [Google Scholar] [Publisher Link]

[2] Yongchao Zhou et al., "Large Language Models are Human-Level Prompt Engineers," *arXiv*, pp. 1-43, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[3] Wayne Xin Zhao et al., "A Survey of Large Language Models," *arXiv*, pp. 1-97, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[4] Ann Yuan et al., "Wordcraft: Story Writing with Large Language Models," *27th International Conference on Intelligent User Interfaces*, pp. 841–852, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[5] Jie Huang, and Kevin Chen-Chuan Chang, "Towards Reasoning in Large Language Models: A Survey," *arXiv*, pp. 1-15, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[6] Blaise Aguera y Arcas, "Do Large Language Models Understand Us?," *Daedalus*, vol. 151, no. 2, pp. 183-197, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[7] Murray Shanahan, "Talking about Large Language Models," *arXiv*, pp. 1-13, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[8] Jiaxin Huang et al., "Large Language Models Can Self-Improve," *arXiv*, pp. 1-19, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[9] Steven T. Piantadosi, and Felix Hill, "Meaning without Reference in Large Language Models," *arXiv*, pp. 1-8, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[10] Matthias C. Rillig et al., "Risks and Benefits of Large Language Models for the Environment," *Environmental Science and Technology*, vol. 57, no. 9, pp. 3464-3466, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[11] Arun James Thirunavukarasu et al., "Large Language Models in Medicine," *Nature Medicine*, vol. 29, pp. 1930–1940, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[12] Yupeng Chang et al., "A Survey on Evaluation of Large Language Models," *arXiv*, pp. 1-42, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Gpt Generative Pretrained Transformer, Almira Osmanovic Thunstrom, and Steinn Steingrimsson, "Can GPT-3 Write an Academic Paper on Itself, with Minimal Human Input?," *HAL Open Science*, pp. 1-8, 2022. [Google Scholar] [Publisher Link]

[14] Fiona Fui-Hoon Naha et al., "Generative AI and ChatGPT: Applications, Challenges, and AI-Human Collaboration," *Journal of Information Technology Case and Application Research*, vol. 25, no. 3, pp. 277–304, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[15] Geoffrey James, "Artifical Intelligence and Document Processing," *Proceedings of the 5th Annual International Conference*, pp. 8-12, 1986. [CrossRef] [Google Scholar] [Publisher Link]

[16] Q. Zhu, and J. Luo, "Generative Pre-Trained Transformer for Design Concept Generation: An Exploration," *Proceedings of the Design Society*, vol. 2, pp. 1825–1834, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[17] Valentinas Gružauskas, and Diwakaran Ragavan, "Robotic Process Automation for Document Processing: A Case Study of a Logistics Service Provider," *Journal of Management*, vol. 36, no. 2, pp. 119-126, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[18] Graham A. Cutting, and Anne-Francoise Cutting-Decelle, "Intelligent Document Processing Methods and Tools in the Real World," *arXiv*, pp. 1-28, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[19] Arman Cohan et al., "Overview of the Third Workshop on Scholarly Document Processing," *Association for Computational Linguistics*, pp. 1-6, 2022. [Google Scholar] [Publisher Link]

[20] Muthu Kumar Chandrasekaran et al., "Overview of the First Workshop on Scholarly Document Processing (SDP)," *Association for Computational Linguistics*, pp. 1–6, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[21] Thomas Hegghammer, "OCR with Tesseract, Amazon Textract, and Google Document AI: a Benchmarking Experiment," *Journal of Computational Social Science*, vol. 5, pp. 861-882, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[22] Dipali Baviskar, Swati Ahirrao, and Ketan Kotecha, "Multi-Layout Unstructured Invoice Documents Dataset: A Dataset for Template-Free Invoice Processing and its Evaluation Using AI Approaches," *IEEE Access*, vol. 9, pp. 101494-101512, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[23] Sk Md Obaidullah et al., *Document Processing Using Machine Learning*, CRC Press, 2019. [Google Scholar] [Publisher Link]

[24] Yuan Y. Tang, Seong-Whan Lee, and Ching Y. Suen, "Automatic Document Processing: A survey," *Pattern Recognit*, vol. 29, no. 12, pp. 1931-1952, 1996. [CrossRef] [Google Scholar] [Publisher Link]

[25] Dipali Baviskar, Swati Ahirrao, and Ketan Kotecha, "A Bibliometric Survey on Cognitive Document Processing," *Library Philosophy and Practice (e-journal)*, pp. 1-31, 2020. [Google Scholar] [Publisher Link]

[26] Tan Yue et al., "DWSA: An Intelligent Document Structural Analysis Model for Information Extraction and Data Mining," *Electronics*, vol. 10, no. 19, pp. 1-16, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Jason T.L. Wang, and Peter A. NG, "Texpros: An Intelligent Document Processing System," *International Journal of Software Engineering and Knowledge Engineering*, vol. 2, no. 2, pp. 171-196, 1992. [CrossRef] [Google Scholar] [Publisher Link]

[28] Xufeng Ling, Ming Gao, and Dong Wang, "Intelligent Document Processing Based on RPA and Machine Learning," *2020 Chinese Automation Congress (CAC), IEEE*, pp. 1349–1353, 2020. [CrossRef] [Google Scholar] [Publisher Link]