Aalto-yliopisto
Aalto-universitet
School of Electrical
Engineering

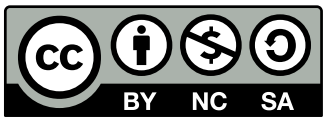
Master's Programme in Data Science

Automating Information Extraction from Non-Standard Financial Reports Using Large Language Models

Enhancing Efficiency through Format-Aware Extraction with Large Language Models

Gabriel Gomes Ziegler

Master's Thesis
2024



Aalto-yliopisto
Aalto-universitet
Aalto University
School of Electrical
Sähkötekniikan
Elektroteknik
Engineering

Author Gabriel Gomes Ziegler

Title Automating Information Extraction from Non-Standard Financial Reports Using Large Language Models — Enhancing Efficiency through Format-Aware Extraction with Large Language Models

Degree programme Data Science

Major ICT Innovation

Supervisor Prof. Bo Zhao

Advisor MS Liliya Shakhpazyan (MSc)

Collaborative partner Datia

Date 21 September 2023 **Number of pages** 40+1 **Language** English

Abstract

The abstract is a short description of the essential contents of the thesis, usually in one paragraph: what was studied and how and what were the main findings.

For a Finnish thesis, the abstract should be written in both Finnish and English; for a Swedish thesis, in Swedish and English. The abstracts for English theses written by Finnish or Swedish speakers should be written in English and either in Finnish or in Swedish, depending on the student's language of basic education. Students educated in languages other than Finnish or Swedish write the abstract only in English. Students may include a second or third abstract in their native language, if they wish.

The abstract text of this thesis is written on the readable abstract page as well as into the pdf file' metadata via the `\thesisabstract` macro (see comment in this TEX file above). Write here the text that goes onto the readable abstract page. You can have special characters, linebreaks, and paragraphs here. Otherwise, this abstract text must be identical to the metadata abstract text.

If your abstract does not contain special characters and it does not require paragraphs, you may take advantage of the `\abstracttext` macro (see the comment in this TEX file below).

Keywords For keywords choose, concepts that are, central to your, thesis

Preface

Thanks notes

Otaniemi, 31 August 2024

Eddie E. Engineer

Contents

1 Introduction

1.1 Structure of the thesis

The thesis is composed by a comprehensive comparison of methods for extracting information from financial reports, with a focus on non-standard reports. The thesis is structured as follows:

1. Introduction (Context, Problem Definition, Objectives)
2. Literature review (Concepts, State of the Art)
3. Methodology (Dataset, Detail how experiments were conducted)
4. Results (Present the results of the experiments)
5. Conclusion (Interpretation of results, implications, limitations)
6. References

1.2 Background of the Field of Study

The field of data extraction from financial reports has evolved significantly with advancements in text processing and machine learning technologies. Historically, this task involved manual data entry or rule-based systems that were labor-intensive and prone to errors. The emergence of Large Language Model (LLM)s, such as Generative Pre-trained Transformer (GPT) and Bidirectional Encoder Representations from Transformers (BERT), has revolutionized this domain. These models have the ability to understand and extract complex financial information from unstructured data, thereby increasing accuracy and efficiency. Recent studies have demonstrated the potential of LLMs in automating financial data extraction, highlighting improvements in processing time and data accuracy over traditional methods.

1.3 General Objective

This study aims to extend the current capabilities of data extraction systems by incorporating advanced LLMs and exploring novel methodologies in the field. The primary goals include: ellaborating a comprehensive comparison of methods for extracting information from financial reports, with a focus on non-standard reports,

enhancing the precision and efficiency of data extraction from financial reports, developing a scalable system capable of processing large volumes of data, and comparing the effectiveness of various LLMs and extraction techniques. By achieving these goals, the study seeks to contribute to the broader understanding of automated data extraction and its application in financial analysis.

1.4 Research Question and Sub-Problems

The primary research question of this study focuses on: "What are the best strategies for using LLMs for more accurate and efficient extraction of financial data from unstructured reports?". Sub-problems in this line of inquiry include: identifying the most effective LLM architectures for financial data recognition, developing methodologies for context-aware data extraction, enhancing the system' ability to handle diverse report formats, and evaluating the impact of training data quality and volume on model performance. These sub-problems are essential for understanding the intricacies of applying LLMs to financial data extraction and for developing a comprehensive solution.

1.5 Scope and Constraints

The scope of this study is limited to the extraction of financial data from English-language reports, focusing on publicly available annual and quarterly financial statements. Key constraints include the variability in report formats, the complexity of financial terminology, and the inherent limitations of current LLM technologies in understanding domain-specific contexts. The study primarily revolves around the use of GPT and BERT models, considering their widespread adoption and state-of-the-art performance in text processing tasks. Main concepts involved include Natural Language Processing (NLP), machine learning, data extraction, and financial analysis, with a particular emphasis on the adaptation and optimization of LLMs for specialized data extraction tasks.

2 Concepts and State of the Art

Documenting and searching for information is an old human practice that can be traced back to as far as 3000 BC, when the Sumerians — the first civilization in the world — used clay tablets with cuneiform inscriptions to keep track of legal documents, transaction records, literature, mythological tales amongst other information. They also created different categories to be able to differentiate tablets from its contents in a classification fashion [1]. Similar practices have remained largely relevant throughout history and with the invention of paper and the printing press, the practice of documenting and storing information evolved allowing for more and more information to be stored and shared physically. Not so long ago, in the 20th century, the invention of the computer and the internet revolutionized the way information is stored and shared, allowing for information to be stored and shared digitally. This allowed for huge amounts of information to be stored and shared in a way that was never possible before. This period in time is often referred to as the information age, also known as the third industrial revolution, which marks a time where information became increasingly accessible and also a commodity, especially later in the 21st century with the rise of Artificial Intelligence (AI) and Machine Learning (ML) models that feed on large datasets to learn and make predictions. Additionally, the creation of Portable Document Format (PDF)s by Adobe in 1993 [2] established a standard for storing and sharing information in a portable format that could be easily shared and printed, which quickly became a standard for sharing information, especially in the business world. The digitalized information stored in PDFs soon became a target for information retrieval and data mining techniques, where systems were developed to extract information from these files based on a variety of approaches, such as heuristic-based methods [3, 4, 5], vector space models [6, 7], probabilistic models [8], and more recently, Deep Learning (DL) models, especially those leveraging the transformer architecture [9, 10, 11, 12].

2.1 Information Retrieval

Information Retrieval (IR) is a field of study that focuses on the organization, storage and retrieval of bibliographic information. IR systems are used to provide a response to a user query with references to documents that contain the information sought by the user. Although the field of IR has been improved and become more popular in recent years with the usage of NLP techniques dominating the field such as Retrieval Augmented Generation (RAG)s, the core task of IR has been studied and applied since the 1940s with pioneers like Vannevar Bush, who, in 1945 envisioned the Memex, a theoretical device that would store extensive collections of documents and allow rapid retrieval and cross-referencing of information [13].

2.2 Document AI

The procedure of extracting information from a document — recently referred to as "Document AI" [14] or "Document Understanding" [15] — is a complex problem

due to the diverse nature of data that PDFs allow to store. Such problem often involves cross-modal interactions where information is represented in both natural language text and visual elements such as tables, charts, and images. In the visual domain, document layout analysis has been widely studied and applied using Computer Vision (CV) techniques to detect and extract elements in the document. Document images processing is usually treated as an object detection task where elements such as text, tables, and images are detected and classified [16, 17].

In many scenarios, the information presented in a document requires an approach that can leverage understanding from both the text and visual domains. For instance, consider tables where the values in cells only have meaning when associated with the header row that are commonly described in the first row or column of the table. Other structures such as charts only convey information when the labels and axes are correctly identified and associated with the data points. These problems are particularly true for financial reports, where information is presented in text, tables, charts, infographics, and other visual elements. The requirement to be able to handle layout invariance is tackled by general-purpose models, such as LLMs, that through pre-training of general-purpose models that learn the position information of the elements as well as the visual information presented with the text such as font size, color, and style. These visual cues can be learned by visual encoders and combined with the text in pre-training stage significantly improving the model's capability to abstract non-standardized information from documents [14].

2.3 Transformers

In recent years, AI systems with DL architectures have been the norm for most of the state-of-the-art models introduced. The first DL models were based on fully-connected neural networks which is composed of layers of neurons that are connected to each other in a feed-forward fashion [18, 19]. Decades after the first fully-connected neural networks were published, the Convolutional Neural Networks (CNN) architecture was proposed by Yann LeCun in 1998 [20] and later popularized by the AlexNet model in 2012 after winning the ImageNet competition [21]. These models represented a significant improvement in the field of CV and were able to learn hierarchical features from images, allowing for more complex patterns to be learned. Meanwhile, in the field of NLP, the Recurrent Neural Networks (RNN) architecture was proposed in 1986 bringing the concept of sequential processing to neural networks allowing previous states to be considered in the processing of the current state [22]. The RNN architecture was able to learn sequential patterns from text data, allowing for the development of models that could generate text, translate languages, and more. However, the RNN architecture had limitations in learning long-range dependencies due to the vanishing gradient problem [23]. To address this issue, the Long Short-Term Memory (LSTM) architecture was proposed in 1997 by Sepp Hochreiter and Jürgen Schmidhuber [24]. The LSTM architecture introduced the concept of memory cells that could store information over time, allowing for the learning of long-range dependencies in sequential data. The next breakthrough in the AI field came with the introduction of the transformer architecture in 2017 that revolutionized the field

of NLP by utilizing self-attention mechanisms to learn intrinsic features [25]. The transformer architecture brought significant improvements in the field of NLP that was demonstrated by the BERT model in 2019 [26] and later by the GPT-3 model in 2020 [27] that would become the state-of-the-art models in the field of NLP. Much inspired by the major advancements in the field of NLP with the transformer architecture, researchers started applying the transformer architecture to the field of CV as an alternative to the long-standing CNN architecture that has been the standard for many years [28]. These models, known as visual transformers, have shown to be competitive with CNNs in many tasks and have demonstrated the potential to learn hierarchical features from images in a more efficient way than CNNs [29].

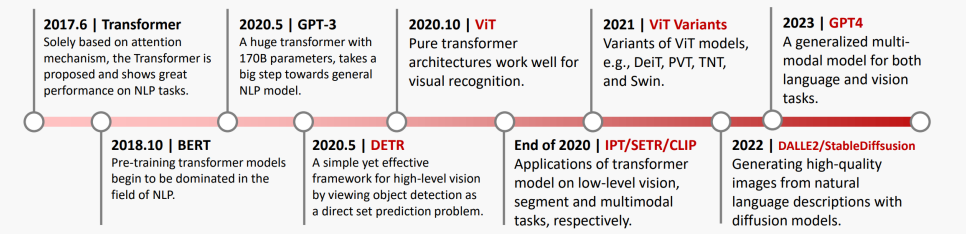


Figure 1: Key milestones in the development of transformer. The vision transformer models are marked in red (Image from 'A Survey on Vision Transformer' [30]).

The original transformer architecture [25] is a sequence-to-sequence model composed of an encoder and a decoder block that are themselves composed of multiple layers of multi-head self-attention mechanisms and feed-forward neural network.

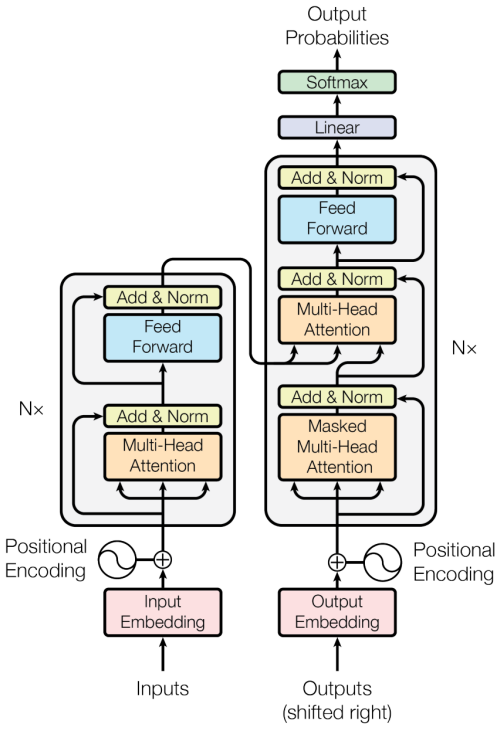


Figure 2: Transformer architecture. The model is composed of an encoder and a decoder block that are themselves composed of multiple layers of multi-head self-attention mechanisms and feed-forward neural network [25].

The attention function, often described as a map of a query and a set of key-value pairs to an output, outputs a weighted sum of the values based on the similarity of the query to the keys computing a compatibility function between the query and the corresponding key.

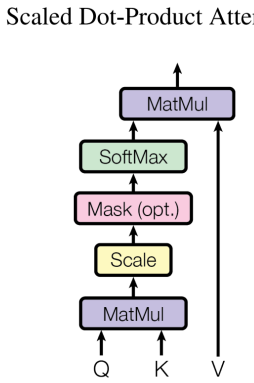


Figure 3: Scaled dot-product attention function [25].

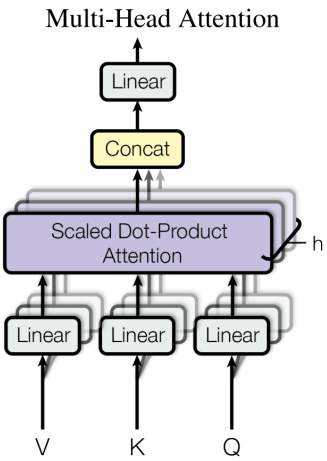


Figure 4: Multi-head attention mechanism [25].

The attention function is computed on a set of queries in a parallel fashion where the queries are packed into a matrix Q, the keys into a matrix K, and the values into a matrix V. The attention function is computed as follows [25]:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

2.3.1 Transformers in Vision

Before the introduction of the transformer architecture in the visual domain, traditional supervised techniques suffered from the necessity of large amounts of labeled data to learn hierarchical features from images limiting the scalability and generalization of such models. The incorporation of models such as BERT, GPT, DALL-E, that are trained on large amounts of text data and generally with a self-supervised learning approach, leads to a decreased dependence on carefully annotated labels, which might allow room for progress on essential cognitive skills that have been challenging in the traditional supervised learning paradigm [31, 32, 33].

The application of transformer architectures to vision tasks has revolutionized the field. Vision Transformers (ViTs) have demonstrated that by treating image patches as sequences of tokens, similar to words in text, one can leverage the powerful sequence modeling capabilities of transformers to process and analyze visual data. ViTs have shown remarkable performance across various tasks, including image classification, object detection, and segmentation, often surpassing the performance of traditional CNNs [29].

With such trend of transformers architectures in the visual domain, OpenAI releases GPT-4 Vision, which integrates advanced transformer architectures specifically designed for understanding and generating visual content. GPT-4 Vision builds on the principles of its text-based equivalent GPT-4 but adapts the transformer model to handle the unique challenges of image inputs [34]. This includes learning complex spatial relationships and capturing fine-grained details from images, which are crucial for tasks such as image synthesis, enhancement, and interpretation

The integration of multimodal learning in GPT-4 Vision allows for processing of textual and visual information, enhancing the model's ability to understand and generate coherent and contextually enriched content. This multimodal capability is particularly valuable in processing documents that contain both text and visual elements, such as financial reports, where information is presented in a variety of formats, including tables, charts, and images. For these reasons and for its availability, GPT-4 Vision is the model of choice for multimodality experiments in this study.

2.4 Large Language Models

LLMs are a class of statistical language models based on neural networks — oftentimes utilizing the transformers architecture — that are large in size and pre-trained on vast amounts of text data. These models represent a significant advancement in the NLP field allowing complex tasks to be performed with high accuracy and efficiency. This

technology has been widely researched and developed by leading big tech companies such as OpenAI, Google, and Meta, [27, 26, 35, 36, 37, 38] that have pushed the boundaries of transformers pretrained models to the point that they can generate human-like text, understand context, and even perform tasks that require domain-specific knowledge. This milestone meant not only a significant improvement in the field of NLP but also a major adoption of artificial intelligence agents by the average user, especially with OpenAI's ChatGPT release in 2022 that delivered a fine-tuned GPT-3.5 model utilizing reinforcement learning from human feedback using the same methods as the InstructGPT model [39].

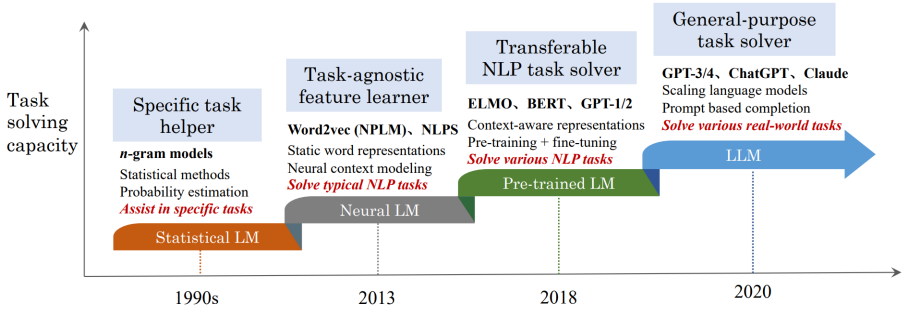


Figure 5: Key milestones of large language models from the task-solving perspective [40].

2.4.1 GPT-4

The fourth GPT release by OpenAI, is a large language model with multilingual and multimodal capabilities that allow it to process image and text inputs, producing text outputs. The model was developed aiming to improve the ability to comprehend and generate natural language text. GPT-4 is often evaluated against human performance on tasks like bar exam, LSAT, SAT, among others, and has shown to be competitive with human performance by achieving top 10% scores on bar exams, compared to GPT-3.5 which achieved bottom 10% scores [36]. The specifics about the model's architecture and training data are not disclosed by OpenAI, but it is known that the model has been pre-trained to predict the next word in a sentence, using publicly available and licensed data fine-tuned with Reinforcement Learning from Human Feedback (RLHF), which has a great impact on the model's performance [36]. While this new model has shown great improvements in language understanding, generation and reduction of hallucinations, it still has limitations in understanding context and generating coherent responses, which is a common issue in large language models [36].

2.4.2 GPT-4V

GPT-4 Vision represents an extension of the capabilities of traditional LLMs into the realm of visual understanding and analysis. By integrating vision-based artificial intelligence technologies with the language processing prowess of GPT-4, this model

can interpret and analyze images, diagrams, and visual data in conjunction with textual information. This multimodal approach enables GPT-4 Vision to perform tasks that require an understanding of both visual and textual content, such as extracting data from charts and graphs in financial reports, identifying key information in documents with complex layouts, and answering questions that depend on visual cues. The development of GPT-4 Vision is a testament to the ongoing advancements in AI, highlighting the move towards more integrated and comprehensive models that can navigate the complexities of human communication and information processing [34].

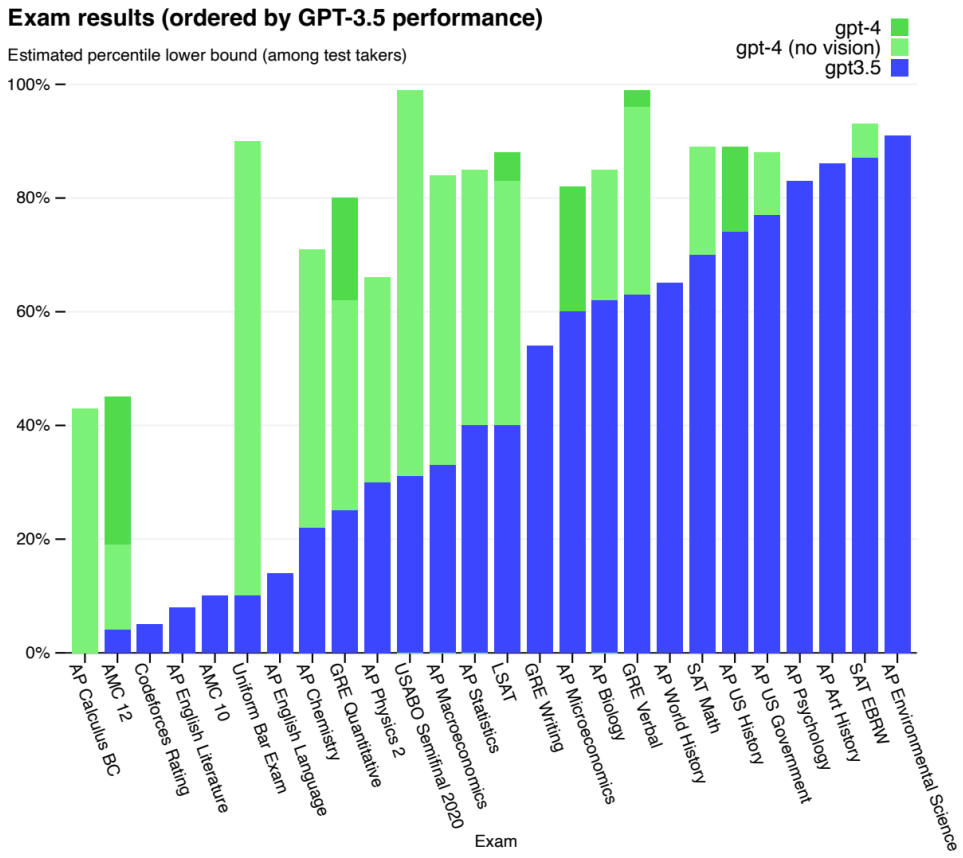


Figure 6

2.5 LLMs for Document AI

LLMs have become a popular strategy in the field of Document AI, transforming how information is extracted, processed, and analyzed from documents. In the context of Document AI, LLMs are utilized to understand the content within documents, ranging from simple text to complex structures like tables and charts, and the relationships between different pieces of information. These models leverage their extensive training on diverse datasets to adapt to the specific challenges posed by document analysis, such as varying formats, layouts, and the integration of multimodal data. Through techniques such as transfer learning and fine-tuning, LLMs can be specialized to perform tasks

including but not limited to information extraction, document summarization, and semantic search within documents. Their ability to process and analyze documents at scale significantly reduces the time and effort required for data entry, extraction, and analysis, enabling more efficient and accurate handling of document-based information [**?**].

2.6 Question answering with RAG

RAG represents a novel approach in leveraging LLMs for the task of question answering. RAG combines the generative capabilities of GPT-like models with retrieval-based methods, which search a large corpus of documents to find relevant information that can aid in generating accurate and informative answers. This technique involves two main components: a retriever, which identifies relevant documents or passages given a query, and a generator, which synthesizes the retrieved information into a coherent response. By integrating these two processes, RAG is able to produce answers that are not only contextually relevant but also enriched with details and insights drawn from a wide range of sources. This method has shown significant promise in improving the accuracy and depth of responses provided by AI systems in question answering applications, particularly in domains where detailed and specific knowledge is required, such as academic research and technical support

2.7 Issues with LLMs for Document AI

2.7.1 Hallucinations

Despite their many advantages, LLMs have known limitations when applied to Document AI tasks. One issue that is introduced with the generative nature of these models is the potential for generating incorrect or misleading information, especially when the input data is ambiguous or incomplete. These mistakes — oftentimes referred to as hallucinations — occur when the generative model create plausible and convincing responses that are incorrect. Although, it is possible to identify and mitigate these so-called hallucinations, studies have shown via learning theory that these mistakes are inherent to the generative nature of LLMs and cannot be completely extinguished [41]. The fact that these mistakes are realistic increases the difficulty of detecting them and bring uncertainty about the reliability of the information provided by the model in a productive environment. Comprehensive studies have been conducted to understand the causes of hallucinations and found that these come from a variety of reasons including noisy data, poor parametric choices, incorrect attention mechanism, improper training procedure, among others. There are two distinct categories of hallucinations identified in the literature: intrinsic hallucination and extrinsic hallucination and they require different strategies to be mitigated [42]. Consider that we have the following source data used as input to an LLM model:

> *The company reported revenues of $1 million in Q1 2022, and $2 million in Q2 2022.*

- **Intrinsic hallucination**: This type of hallucination occurs when the model generates information that contradicts the input data. A case of intrinsic hallucination would be if the model generated the following output:

  > *The company reported revenues of $1 million in Q1 2022, and $3 million in Q2 2022.*

  Here, the model has generated information that is inconsistent with the input data since the revenue reported in Q2 2022 is incorrect according to the source data. This type of hallucination can be particularly problematic in document analysis and is the one that this study dives the most into.

- **Extrinsic hallucination**: Extrinsic hallucination occurs when the model generates incorrect information that is not present in the input data but is plausible given the context. For example, if the model generated the following output:

  > *The company is projected to report revenues of $3 million in Q3 2022.*

  This information is not present in the input data and cannot be inferred from the given context, therefore it is classified as an extrinsic hallucination.

There are several techniques that aim in mitigating these issues, such as using retrieval-based methods [43], fine-tuning a model on a specific domain [27] or by retrying the generation process multiple times while validating the output against a defined model such as the tools *LangChain* [44] and *instructor* [45] propose. In this study, we show applications of how using defined models with *pydantic* [46] and *instructor* can help in mitigating hallucinations in the context of document data extraction.

2.7.2 Interpretability and Explainability

Ever since DL models gained popularity in productive environments, many concerns have been raised regarding the interpretability and explainability of these models, particularly in high-stakes applications such as healthcare, finance, and law where accountability and transparency are crucial and these models can have societal impacts, e.g. inequity, discrimination, misuse, economic and environmental impact, ethical concerns, among others. As already demonstrated in several studies, DL models are often treated as black boxes, because of the complex and non-linear nature of their architectures, making it difficult to understand how they arrive at their decisions and generate their outputs [47, 48]. LLMs are no exception to this, as their complex attention mechasnisms and deep learning architectures make it challenging to interpret the reasoning behind their predictions and the information they generate.

The lack of interpretability and explainability can be a significant barrier to the adoption of LLMs in critical applications, as it not only raises concerns about the reliability, trustworthiness, and ethical implications of the decisions made by these models, but also increases complexity when debugging and improving the models.

The uncertainty involving LLMs outputs poses a challenge for users who need to understand what kinds of inputs lead to incorrect outputs. This is particularly important in the context of this study as the information extracted from financial reports is used to make critical business decisions, and the reliability and accuracy of the extracted data are paramount. For these reasons, we investigate nuances on the documents that lead to errouneous predictions so that they can be avoided in the future.

3 Financial Reports Dataset

The dataset used in this study consists of a collection of annual and quarterly reports from various public companies across different industries and different countries. We count with a total of 1000 reports, carefully annotated manually with the Greenhouse Gas (GHG) emissions data present in the reports. These reports can contain a wide range of information, including financial statements, management discussions, and analysis, auditor reports, and of course, Environmental, Social, and Governance (ESG) disclosures including GHG emissions data, which is the topic of interest in this study. The documents are stored in PDF format with varying layouts, fonts, structures and number of pages.

We opted to create a diverse dataset having heterogeneous formats to report GHG emission data, including text, tables, charts and images to evaluate the effectiveness of different strategies for extracting information from financial reports as this represents a realistic nature of the data that the system would encounter in a real-world scenario in productive environments. Below we present a few samples to illustrate the diversity of formats and layouts used to report GHG emissions data in financial reports.

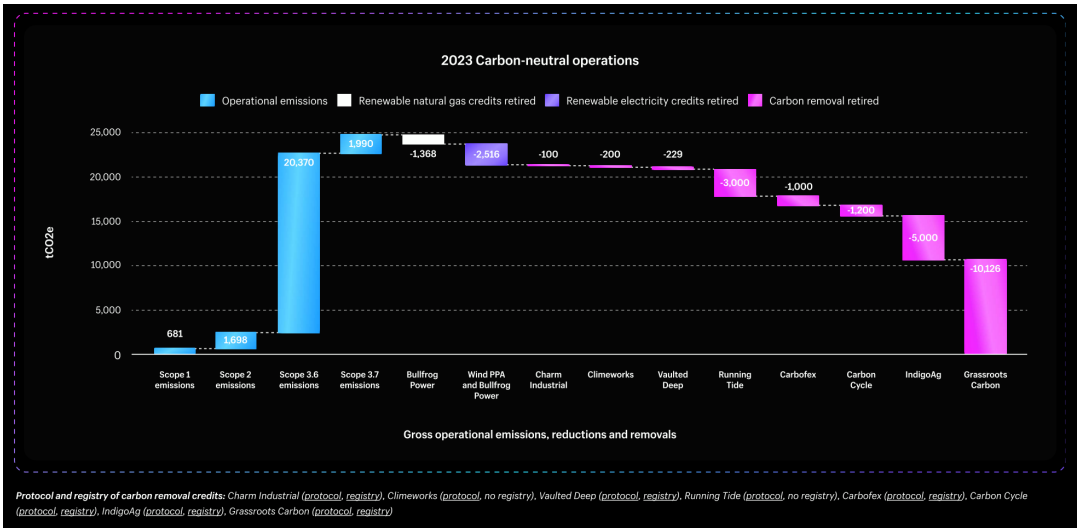


Figure 7: Document reporting GHG emissions data using a bar chart figure.

The figure above shows a case where the report presents the GHG emissions data using a bar chart figure. Although this is visually rich and easy to comprehend for the human eye, it poses a huge challenge for an Optical Character Recognition (OCR) system to extract the data from the chart. The text may be detected and extracted correctly by the OCR system, but the context of the data will likely be lost, leading to an extraction with high uncertainty. A visual or multimodal approach would be necessary to extract the data from this type of document.

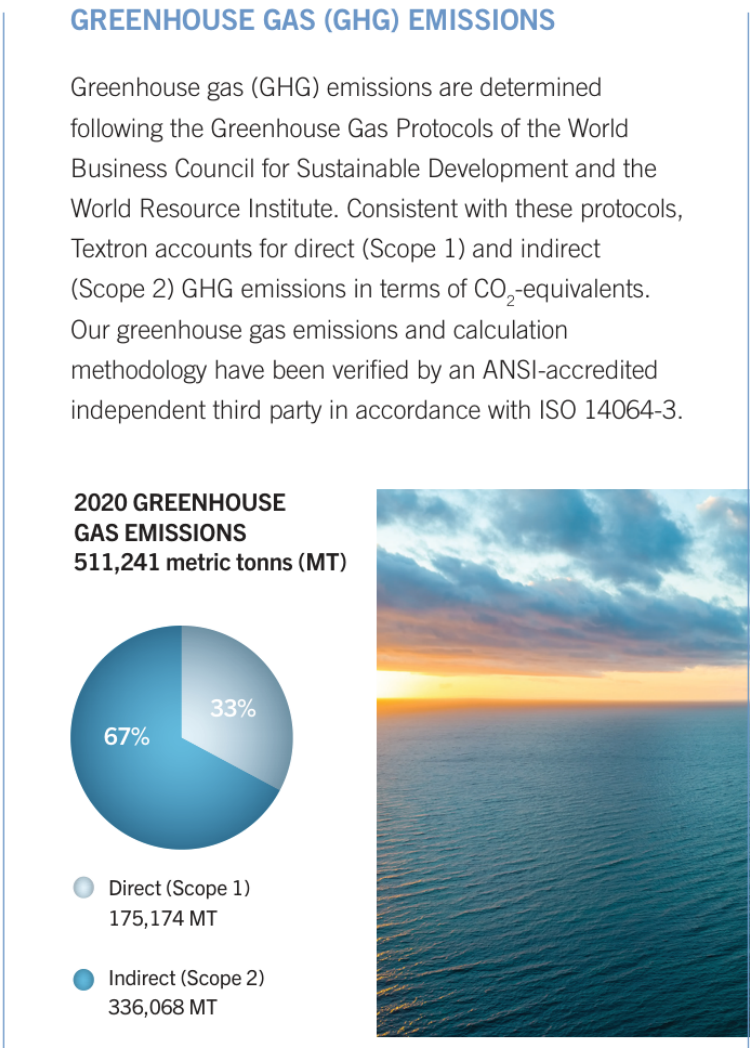
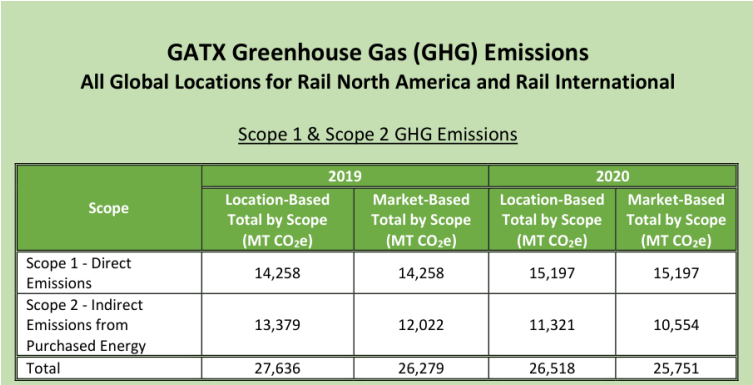


Figure 8: Document reporting GHG emissions data in image with a pie chart.

Similarly to figure 7, the figure above shows a case where the report presents the GHG emissions data using a pie chart figure. In this case, since we would only be interested in the values presented in the text under the image, the OCR-based approach could lead to a successful extraction of the data, however this is another example to display the diversity of formats that companies use to report key information in their financial reports.



GATX Greenhouse Gas (GHG) Emissions
All Global Locations for Rail North America and Rail International

Scope 1 & Scope 2 GHG Emissions

Scope	2019		2020	
	Location-Based Total by Scope (MT CO_2e)	Market-Based Total by Scope (MT CO_2e)	Location-Based Total by Scope (MT CO_2e)	Market-Based Total by Scope (MT CO_2e)
Scope 1 - Direct Emissions	14,258	14,258	15,197	15,197
Scope 2 - Indirect Emissions from Purchased Energy	13,379	12,022	11,321	10,554
Total	27,636	26,279	26,518	25,751

Figure 9: Document reporting GHG emissions data in a table.

One of the most common layouts present in the dataset is the table format. It is a great way to present structured data over a period of time in a concise, objective and easy-to-read way. This is also one of the best formats for an OCR-based approach to extract the data, as the data is already structured and there is a less chances of having context loss. However, OCRs are not error-free with tables, especially in cases where the table is structured in a non-conventional way, such as merged cells, rotated text, or cells with no borders. Additionally, Document AI researchers have extensively studied and developed models to detect, extract and understand tables in documents [49, 3, 17, 10, 50, 51].

3.1 Additional data intricacies relevant to our problem

TODO check if these should be in experiment

3.1.1 Unit of measurement

The diversity in such documents, is not only present in the layouts, but also in the metrics reported. Most companies have been seen to disclose their GHG emissions data in metric tons of CO_2 equivalent, however, other companies have report their emissions in other units such as kilograms of CO_2, metric tons of CO_2 per unit of production, and some companies that have very large numbers report their emissions in million metric tons of CO_2 equivalent. Since the unit of measurement is crucial for the correct interpretation of the data, it is important to also be able to extract this information from the reports. Therefore, if our system correctly detects the values in the document, but fail to output a normalized unit of measurement — in our case, we use metric tonnes — the system output will be considered incorrect.

3.1.2 Scope 2 emissions origin

The GHG Scope 2 emissions are divided into three categories: location-based, market-based, and undefined. The location-based emissions are the emissions calculated based on the location of the company's operations, the market-based emissions are the

emissions calculated based on the market where the company sells its products, and the undefined emissions are the emissions that are not clearly defined as location-based or market-based. It is important to be able to extract this information from the reports, as it is crucial for the correct interpretation of the data. Therefore, if our system correctly detects the values in the document, but fail to output the correct category for the Scope 2 emissions, the system output will be considered incorrect.

4 Strategies for information extraction from financial reports

Different strategies for extracting information from financial reports have been implemented in order to compare their effectiveness across diverse report formats and content types. The study brings a strategy that is focused solely on text information, an image-analysis approach that extracts information from images contained in the reports, and a multimodal approach that combines image and text information to validate extracted data from more than one source of truth. For these different experiments, we consider that the core engine changes, but we maintain the same pre-processing steps across them to ensure that the comparison only takes into account the core feature of extracting data from a given input.

4.1 System Specifications

The experiments proposed in this study were conducted on a machine with the following specifications:

- **CPU**: AMD Ryzen 7 3700X 16 threads at 3.600 GHz
- **Memory**: 32GB at 3200 MHz
- **Operating System**: Linux Manjaro 6.1.55-1
- **Python**: 3.11.5

4.2 Experiments definition

For the sake of setting up an OCR challenge that that is relevant to business cases and provides the opportunity to compare different strategies for extracting information from financial reports, we establish a simple set of indicators of interest, a desired schema for the extracted data, and a set of metrics to evaluate the performance of the different strategies.

4.2.1 Indicators of interest

Given Datia's strong presence in the ESG domain, we have chosen to focus on a set of ESG indicators that are commonly reported in financial statements. Therefore, this challenge focuses on correctly extracting values for the following indicators:

- **GHG Scope 1 emissions**: The total amount of GHG emissions directly produced by a company.
- **GHG Scope 2 emissions**: The total amount of GHG emissions indirectly produced by a company.

  - **Location-based**: Emissions calculated based on the location of the company's operations.
  - **Market-based**: Emissions calculated based on the market where the company sells its products.
  - **Undefined**: Emissions that are not clearly defined as location-based or market-based.

- **GHG Scope 3 emissions**: The total amount of GHG emissions produced by in the value chain of a company. Scope 3 emissions can also be broken down into 3.1, 3.3, 3.4, 3.6, 3.7 and 3.11 categories, however, for the purpose of this study, we will only consider the total Scope 3 emissions.

- **Reported unit**: The unit of measurement used for the emissions.

These are crucial indicators for assessing a company's environmental impact and sustainability practices, and they are often reported in financial statements as part of the company's ESG disclosures. It's also important to be able to extract the unit of measurement used for the emissions, because different companies may report their emissions in different units, such as metric tons of CO_2 equivalent, kilograms of CO_2, or other units.

4.3 Extracted data schema

The extracted data from the financial reports should follow a specific schema to ensure consistency and comparability across different strategies. Since most of the strategies are LLM-based, defining a schema adds complexity to the system, as hallucination could lead to correct data being extracted but in the wrong format. For these reasons, we define a schema that is simple and straightforward, focusing on the key indicators of interest and their values for each year reported. Here is an example of the JavaScript Object Notation (JSON) schema for the extracted data:

```
{
    "metrics": {
        "2022": {
            "scope_1": 88200000.0,
            "scope_2": {
                "location_based": null,
                "market_based": null,
                "undefined": 200000.0
            },
            "scope_3": 38800000.0
        },
        "2023": {
            "scope_1": 75100000.0,
            "scope_2": {
```

```
                "location_based": null,
                "market_based": null,
                "undefined": 200000.0
            },
            "scope_3": 36600000.0
        }
    },
    "extracted_pages": [1, 4, 10],
    "reported_unit": "million_metric_tonnes",
}
```

This schema guarantees that the extracted data can be processed by the system and also be represented in the same unit of measurement, ensuring that the comparison between the different strategies is fair and accurate. The `extracted_pages` field is used to store the page numbers from which the data was extracted, allowing for traceability and validation of the extracted information.

4.4 Evaluation Criteria

To thoroughly assess the performance of the proposed systems for extracting indicators from financial reports, a combination of quantitative and qualitative metrics is employed. These metrics are designed to measure both the accuracy of the extracted data and the robustness of the extraction process against various types of errors. Here, we delineate the key metrics and evaluation criteria used.

4.4.1 Precision and Recall

Precision and Recall are critical metrics for evaluating the effectiveness of the data extraction system:

- **Precision** assesses the proportion of data points extracted by the model that are correct and relevant. A high precision rate indicates fewer instances of fabricated metrics and irrelevant data extraction.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{2}$$

- **Recall** measures the system's ability to retrieve all relevant data points from the document. High recall is essential to ensure no significant data is missed.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \tag{3}$$

4.4.2 MAPE

For numerical errors, the Mean Absolute Percentage Error (MAPE) is used to quantify the accuracy of the extracted values. This metric calculates the average percentage difference between the extracted values and the ground truth values, providing insights into the model's performance in capturing the correct numerical data while accounting for scale and magnitude differences.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \times 100 \tag{4}$$

4.4.3 Error Types Analysis

In addition to quantitative metrics, an error analysis is conducted to identify the types of errors made by the system during the extraction process.

- **Missing Data**: Indicates data that should have been extracted but was not. This error impacts the `recall` metric.
- **Incorrect Values**: Reflects errors in the value of the data extracted. These are numerical errors that impact the `MAPE` metric.
- **Fabricated Metrics**: Metrics that are not present in the document but appear in the extracted output — also referred to as extrinsic hallucination. This error impacts `precision`.

4.5 Common processing steps

For all of the experiments, the system receives a PDF as input and common pre and post-processing steps are applied to the data to ensure that the system is able to extract the information from the reports.

4.5.1 Pre-processing: finding pages of interest

A common pre-processing step is to find the pages of interest — those that might contain the information that we are looking for — in the PDF document.

```
def extract_data_from_pdf(pdf_file: str):
  # Load the PDF file
  doc = fitz.open(pdf_file)

  for page in doc:
    # extract text from page
    text = page.get_text("text")

    # function that will look for matches
```

```
# for the indicators of interest
matches = search_for_keywords(text)

# this page does not contain
# any relevant information
if matches is None:
    return

# Then different strategies follow...
```

This pre-processing step is used to iterate over the pages in the PDF document and find the pages of interest so that the core engine is only applied to these pages. This is an important step to reduce the processing time and to avoid unnecessary processing of pages that do not contain relevant information, however it is also crucial to make sure that no false negatives are generated, as this could lead to missing pages of interest.

4.5.2 Parsing LLMs outputs

As of the time of writing, the text GPT-4 model has a parameter flag to determine the output format of the model while the vision model does not have this feature. This means that for text, it is possible to specify JSON as the output format, however for the vision model, the output is always a string and that leads to uncertainty whether the output string can be parsed into a JSON format by the system. Additionally, the parameter flag does not guarantee that the output JSON will contain the correct keys and meet the desired schema. For these reasons, it is important to have a parsing step that validates the output of the LLM models and ensures that the extracted data is in the expected format.

This is where the *pydantic* and *instructor* library comes into play, as we can use *pydantic* to define the model schema and validate fields and types, and *instructor* will handle any imperfections in the model by retrying the generation process n times and validating the output against the defined schema.

4.5.3 Post-processing: consolidating information from different pages

In an ideal world, all the information that we are looking for would be contained in one and only one page of the PDF document and our system would be able to find the page, extract and return. However, this is not the case in practice, as different companies choose different designs to display their information and sometimes the indicators are spread across different pages, or even repeated in different sections of the report. Since our system needs to be able to resolve conflicts in the scenario of having multiple pages with the same information (potentially with different values due to different local regulations, units, and so forth), a strategy for consolidating the information is needed. After testing out different strategies like frequency bags, picking the output with most found values, and LLMs, we have decided to stick with the LLM-approach by using GPT-4 model, to consolidate the information from different pages. Here is a demonstration of how this can be done:

```python
def consolidate_data(data: list[str]) -> str:
  """
  Consolidate data from different pages

  Args:
    data: List of JSON strings containing the data
          extracted from different pages

  Returns:
    consolidated_data: Unique JSON string containing
                       the consolidated data
  """
  # Send request with data and instructions to GPT4
  consolidated_data = openai.chat.completions.create(
    model="gpt-4-turbo-preview",
      response_format={"type": "json"},
      messages=[
        {
          "role": "system",
          "content": """Consolidate the following data
                        into only one JSON string..."""",
        },
        {"role": "user", "content": data},
      ]
    )
  return consolidated_data
```

This is a simple example of how the GPT-4 model can be used to consolidate information from different pages of the PDF document and can definitely be improved by adding more context and validation steps, such as checking for inconsistencies in the data, before returning the consolidated information.

4.6 Text-only approach with LLMs

In this experiment, we use the GPT-4 architecture, more especifically *gpt-4-0125-preview* released in December 2023, to extract information from the text contained in the financial reports without considering any images or visual data. Evidently, this approach is limited to the information that is present in the text and does not take into account any charts or important visual cues that the report might contain, thus it is expected that this system may fail for cases where the indicator are presented in a visual form.

4.7 Image-only approach with LLMs

In this experiment, we use the GPT-4 Vision model to extract information from the images contained in the financial reports without considering any text data. Since OpenAI's visual model is highly accurate in extracting information from images, this approach is expected to have an overall performance superior to the text-only approach because it will be able to generalize the information from images but it is also expected to have an increased proclivity to hallucinations, as the model is not able to validate the information extracted from the images with the text data.

For this experiment, after the preprocessing step, for each of the pages of interest, the system applies the following image-processing transforms:

- Dots Per Inch (DPI): The image is set to 300 DPI to ensure that the model can extract the information with high quality. This setting has shown to be one of the optimal settings for OCR tasks [52].
- Grayscale: The image is converted to grayscale to reduce the amount of information that the model needs to process.
- Downscaling: The image is downscaled such that they fit OpenAI's GPT-4 Vision criteria:

  > *...images are first scaled to fit within a 2048 x 2048 square, maintaining their aspect ratio. Then, they are scaled such that the shortest side of the image is 768px long [53].*

- Compression: The image is compressed to a 90% quality to significantly reduce the size of the image while keeping sufficient quality for image analysis.

These transforms have the goal of reducing the amount of information that the model needs to process, while maintaining the quality of the information extracted from the images. This is important not only for model performance, but also to keep the number of tokens processed lower, reducing the cost of the operation.

The system sends a request to OpenAI's GPT-4 Vision model with the preprocessed image and the instructions where the model is asked to extract the key indicators from the image into a predefined JSON schema. This is necessary and important because not only does the model need to be able to extract the information from the images, but it also needs to produce the information in a format that can be parsed by the system, otherwise it might raise unexpected errors on the client side.

4.8 Multimodal LLMs to extract information from images and text

In this experiment, we use a multimodal approach to extract information from the financial reports, leveraging information from both textual and visual domains. This approach combines the strengths of the text-only and image-only methods while

mitigating their respective weaknesses by validating the information extracted from the images with the text data and vice versa.

The system follows the same pre-processing steps as the image-only approach before 4.7 sending the image and instructions to the GPT-4 Vision model. The difference for this experiment is that the system also extracts the text from the page and creates a set of all the numbers mentioned in the page, which will be used to validate if the information given by the model is consistent with the text data or if it is an extrinsic hallucination.

In order to mitigate intrinsic and extrinsic hallucinations, we implement a simple lookup method that will search for the closest value in the text data to the one extracted from the image that is within a threshold limit so that the values that are within the threshold are considered valids and are readjusted while the ones that are not are considered hallucinations and are discarded.

This system is expected to have the best performance among the three approaches, as it is able to validate the information extracted from the images with the text data and vice versa, however it is also expected to have the highest processing time due to the increased complexity of the system.

5 Results

TODO: describe the results of each experiment given the defined metrics for evaluation

6 Conclusions

TODO: add conclusion section

References

[1] I. Finkel and J. Taylor, *Cuneiform*, ser. Ancient scripts. J. Paul Getty Museum, 2015. ISBN 9781606064474. [Online]. Available: https://books.google.com.br/books?id=cf7NrQEACAAJ

[2] Adobe Systems Incorporated, "Pdf timeline," https://www.adobe.com/acrobat/resources/pdf-timeline.html, 2023, accessed: 2024-04-20.

[3] D. Pinto, A. McCallum, X. Wei, and W. B. Croft, "Table extraction using conditional random fields," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, ser. SIGIR '03. New York, NY, USA: Association for Computing Machinery, 2003. doi: 10.1145/860435.860479. ISBN 1581136463 p. 235–242. [Online]. Available: https://doi.org/10.1145/860435.860479

[4] F. Peng and A. McCallum, "Information extraction from research papers using conditional random fields," *Information Processing & Management*, vol. 42, no. 4, pp. 963–979, 2006. doi: https://doi.org/10.1016/j.ipm.2005.09.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457305001172

[5] H. Fang, T. Tao, and C. Zhai, "A formal study of information retrieval heuristics," in *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '04. New York, NY, USA: Association for Computing Machinery, 2004. doi: 10.1145/1008992.1009004. ISBN 1581138814 p. 49–56. [Online]. Available: https://doi.org/10.1145/1008992.1009004

[6] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, p. 613–620, nov 1975. doi: 10.1145/361219.361220. [Online]. Available: https://doi.org/10.1145/361219.361220

[7] S. K. Wong, W. Ziarko, V. V. Raghavan, and P. C. Wong, "On modeling of information retrieval concepts in vector spaces," *ACM Trans. Database Syst.*, vol. 12, no. 2, p. 299–321, jun 1987. doi: 10.1145/22952.22957. [Online]. Available: https://doi.org/10.1145/22952.22957

[8] M. E. Maron and J. L. Kuhns, "On relevance, probabilistic indexing and information retrieval," *J. ACM*, vol. 7, no. 3, p. 216–244, jul 1960. doi: 10.1145/321033.321035. [Online]. Available: https://doi.org/10.1145/321033.321035

[9] Z. Wang, Y. Xu, L. Cui, J. Shang, and F. Wei, "Layoutreader: Pre-training of text and layout for reading order detection," 8 2021. [Online]. Available: http://arxiv.org/abs/2108.11591

[10] J. Li, Y. Xu, T. Lv, L. Cui, C. Zhang, and F. Wei, "Dit: Self-supervised pre-training for document image transformer," in *Proceedings of the 30th ACM International Conference on Multimedia*, ser. MM '22. New York, NY, USA: Association for Computing Machinery, 2022. doi: 10.1145/3503161.3547911. ISBN 9781450392037 p. 3530–3539. [Online]. Available: https://doi.org/10.1145/3503161.3547911

[11] M. Li, T. Lv, J. Chen, L. Cui, Y. Lu, D. Florencio, C. Zhang, Z. Li, and F. Wei, "Trocr: transformer-based optical character recognition with pre-trained models," in *Proceedings of the Thirty-Seventh AAAI Conference on Artificial Intelligence and Thirty-Fifth Conference on Innovative Applications of Artificial Intelligence and Thirteenth Symposium on Educational Advances in Artificial Intelligence*, ser. AAAI'23/IAAI'23/EAAI'23. AAAI Press, 2023. doi: 10.1609/aaai.v37i11.26538. ISBN 978-1-57735-880-0. [Online]. Available: https://doi.org/10.1609/aaai.v37i11.26538

[12] Y. Xu, M. Li, L. Cui, S. Huang, F. Wei, and M. Zhou, "Layoutlm: Pre-training of text and layout for document image understanding." Association for Computing Machinery, 8 2020. doi: 10.1145/3394486.3403172. ISBN 9781450379984 pp. 1192–1200.

[13] V. Bush, "As We May Think," *Atlantic Monthly*, vol. 176, no. 1, pp. 641–649, March 1945. doi: 10.1145/227181.227186. [Online]. Available: http://www.theatlantic.com/doc/194507/bush

[14] L. Cui, Y. Xu, T. Lv, and F. Wei, "Document ai: Benchmarks, models and applications," 11 2021. [Online]. Available: http://arxiv.org/abs/2111.08609

[15] N. Subramani, A. Matton, M. Greaves, and A. Lam, "A survey of deep learning approaches for OCR and document understanding," *CoRR*, vol. abs/2011.13534, 2020. [Online]. Available: https://arxiv.org/abs/2011.13534

[16] X. Yang, E. Yumer, P. Asente, M. Kraley, D. Kifer, and C. L. Giles, "Learning to extract semantic structure from documents using multimodal fully convolutional neural networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. doi: 10.1109/CVPR.2017.462 pp. 4342–4351.

[17] S. Schreiber, S. Agne, I. Wolf, A. Dengel, and S. Ahmed, "Deepdesrt: Deep learning for detection and structure recognition of tables in document images," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017. doi: 10.1109/ICDAR.2017.192 pp. 1162–1167.

[18] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," in *Psychological Review*, ser. PR '58. Cambridge, MA, USA: MIT Press, 1958. ISBN 9780262181114 pp. 386–408.

[19] ——, "Principles of neurodynamics: Perceptrons and the theory of brain mechanisms," in *Neural Networks*, ser. NN '61. Cambridge, MA, USA: MIT Press, 1961. ISBN 9780262181114 pp. 386–408.

[20] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," in *Proceedings of the IEEE*, ser. IEEE '98. Washington, DC, USA: IEEE, 1998. ISBN 1558605529 pp. 2278–2324.

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, ser. NIPS '12. Red Hook, NY, USA: Curran Associates Inc., 2012. ISBN 9781450319895 pp. 1097–1105.

[22] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[23] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen Netzen," *Diploma Thesis*, July 1991. [Online]. Available: http://www.idsia.ch/~juergen/SeppHochreiter1991ThesisAdvisorSchmidhuber.pdf

[24] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997. doi: 10.1162/neco.1997.9.8.1735. [Online]. Available: https://doi.org/10.1162/neco.1997.9.8.1735

[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017. ISBN 9781510860964 p. 6000–6010.

[26] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4171–4186, 2019.

[27] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[28] Y.-C. Chen, Y.-S. Liu, Z. Kira, and G. AlRegib, "Pre-trained image processing transformer," *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3530–3539, 2021.

[29] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3530–3539, 2021.

[30] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo, Z. Liu, Y. Tang, A. Xiao, C. Xu, Y. Xu, Z. Yang, Y. Zhang, and D. Tao, "A survey on vision transformer," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2023. doi: 10.1109/TPAMI.2022.3152247

[31] R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, M. S. Bernstein, J. Bohg, A. Bosselut, E. Brunskill, E. Brynjolfsson, S. Buch, D. Card, R. Castellon, N. S. Chatterji, A. S. Chen, K. A. Creel, J. Davis, D. Demszky, C. Donahue, M. Doumbouya, E. Durmus, S. Ermon, J. Etchemendy, K. Ethayarajh, L. Fei-Fei, C. Finn, T. Gale, L. E. Gillespie, K. Goel, N. D. Goodman, S. Grossman, N. Guha, T. Hashimoto, P. Henderson, J. Hewitt, D. E. Ho, J. Hong, K. Hsu, J. Huang, T. F. Icard, S. Jain, D. Jurafsky, P. Kalluri, S. Karamcheti, G. Keeling, F. Khani, O. Khattab, P. W. Koh, M. S. Krass, R. Krishna, R. Kuditipudi, A. Kumar, F. Ladhak, M. Lee, T. Lee, J. Leskovec, I. Levent, X. L. Li, X. Li, T. Ma, A. Malik, C. D. Manning, S. P. Mirchandani, E. Mitchell, Z. Munyikwa, S. Nair, A. Narayan, D. Narayanan, B. Newman, A. Nie, J. C. Niebles, H. Nilforoshan, J. F. Nyarko, G. Ogut, L. Orr, I. Papadimitriou, J. S. Park, C. Piech, E. Portelance, C. Potts, A. Raghunathan, R. Reich, H. Ren, F. Rong, Y. H. Roohani, C. Ruiz, J. Ryan, C. R'e, D. Sadigh, S. Sagawa, K. Santhanam, A. Shih, K. P. Srinivasan, A. Tamkin, R. Taori, A. W. Thomas, F. Tramèr, R. E. Wang, W. Wang, B. Wu, J. Wu, Y. Wu, S. M. Xie, M. Yasunaga, J. You, M. A. Zaharia, M. Zhang, T. Zhang, X. Zhang, Y. Zhang, L. Zheng, K. Zhou, and P. Liang, "On the opportunities and risks of foundation models," *ArXiv*, 2021. [Online]. Available: https://crfm.stanford.edu/assets/report.pdf

[32] R. Zellers, Y. Bisk, A. Farhadi, and Y. Choi, "From recognition to cognition: Visual commonsense reasoning," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. doi: 10.1109/CVPR.2019.00688 pp. 6713–6724.

[33] R. Martín-Martín, M. Patel, H. Rezatofighi, and et al., "JRDB: A Dataset and Benchmark of Egocentric Robot Visual Perception of Humans in Built Environments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 6748–6765, 2023. doi: 10.1109/TPAMI.2021.3070543

[34] OpenAI, "Gpt-4v(ision) system card," 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:263218031

[35] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, E. Hesse, H. Wang, J. Dorado, M. Park, J. Foo, E. Steorts, and I. Sutskever, "Learning transferable visual models

from natural language supervision," *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 3530–3539, 2021.

[36] OpenAI, "Gpt-4 technical report," 3 2023. [Online]. Available: http://arxiv.org/abs/2303.08774

[37] G. T. et al., "Gemini: A family of highly capable multimodal models," 2024. [Online]. Available: https://arxiv.org/abs/2312.11805

[38] ——, "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context," 2024. [Online]. Available: https://arxiv.org/abs/2403.05530

[39] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe, "Training language models to follow instructions with human feedback," 2022. [Online]. Available: https://arxiv.org/abs/2203.02155

[40] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li, X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, "A survey of large language models," 2023. [Online]. Available: https://arxiv.org/abs/2303.18223

[41] Z. Xu, S. Jain, and M. Kankanhalli, "Hallucination is inevitable: An innate limitation of large language models," 2024.

[42] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Comput. Surv.*, vol. 55, no. 12, mar 2023. doi: 10.1145/3571730. [Online]. Available: https://doi.org/10.1145/3571730

[43] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020. ISBN 9781713829546

[44] H. Chase, "LangChain," Oct. 2022. [Online]. Available: https://github.com/langchain-ai/langchain

[45] J. Liu, "Instructor: Structured LLM Outputs," Jun. 2023. [Online]. Available: https://github.com/jxnl/instructor

[46] S. Colvin, E. Jolibois, H. Ramezani, A. G. Badaracco, T. Dorsey, D. Montague, S. Matveenko, M. Trylesinski, S. Runkle, D. Hewitt, and A. Hall, "Pydantic," 3 2024. [Online]. Available: https://docs.pydantic.dev/latest/

[47] D. Castelvecchi, "Can we open the black box of ai?" *Nature News*, vol. 538, no. 7623, p. 20, 2016.

[48] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, aug 2018. doi: 10.1145/3236009. [Online]. Available: https://doi.org/10.1145/3236009

[49] S. Paliwal, V. D, R. Rahul, M. Sharma, and L. Vig, "Tablenet: Deep learning model for end-to-end table detection and tabular data extraction from scanned document images," 2020. [Online]. Available: https://arxiv.org/abs/2001.01469

[50] A. Gilani, S. R. Qasim, I. Malik, and F. Shafait, "Table detection using deep learning," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 01, 2017. doi: 10.1109/ICDAR.2017.131 pp. 771–776.

[51] F. Shafait and R. Smith, "Table detection in heterogeneous documents," in *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, ser. DAS '10. New York, NY, USA: Association for Computing Machinery, 2010. doi: 10.1145/1815330.1815339. ISBN 9781605587738 p. 65–72. [Online]. Available: https://doi.org/10.1145/1815330.1815339

[52] W. Bieniecki, S. Grabowski, and W. Rozenberg, "Image preprocessing for improving ocr accuracy," in *2007 International Conference on Perspective Technologies and Methods in MEMS Design*, 2007. doi: 10.1109/MEMSTECH.2007.4283429 pp. 75–80.

[53] "Openai vision api documentation," https://platform.openai.com/docs/guides/vision, OpenAI, 2024, accessed: 2024-04-11.

A Contents of an appendix