Aalto-yliopisto
Sähkötekniikan
korkeakoulu

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
Sähkötekniikan
korkeakoulu

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** 28+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

Aalto-yliopisto
Sähkötekniikan
korkeakoulu

Tekijä Gabriel Gomes Ziegler

Työn nimi Opinnäyteen otsikko — Opinnäytteen mahdollinen alaotsikko

Koulutusohjelma Elektroniikka ja sähkötekniikka

Pääaine Sopiva pääaine

Työn valvoja Prof. Pirjo Professori

Työn ohjaajat TkT Alan Advisor, DI Elsa Expert

Yhteistyötaho Yhtiön tai laitoksen nimi (tarvittaessa)

Päivämäärä 21.9.2023 **Sivumäärä** 28+1 **Kieli** englanti

Tiivistelmä

Tiivistelmä on lyhyt kuvaus työn keskeisestä sisällöstä usein yhtenä kappaleena: mitä tutkittiin ja miten sekä mitkä olivat tärkeimmät tulokset. Suomenkielisen opinnäytteen tiivistelmä kirjoitetaan suomeksi ja englanniksi ja ruotsinkielisen vastaavasti ruotsiksi ja englanniksi. Suomen- tai ruotsinkielisten opiskelijoiden, joiden opinnäytteen kieli on englanti, tulee kirjoittaa tiivistelmänsä englanniksi ja koulusivistyskielellään. Muiden kuin koulusivistyskieleltään suomen- tai ruotsinkielisten tulee kirjoittaa tiivistelmänsä vain englanniksi. Opiskelija voi halutessaan lisätä opinnäytteeseensä toisen tai kolmannen tiivistelmän omalla äidinkielellään. Tämän opinnäytteen tiivistelmäteksti kirjoitetaan opinnäytteen luettavan osan lomakkeen lisäksi myös pdf-tiedoston metadataan. Kirjoita tähän metadataan kirjoitettavaa teksti. Metadatatekstissa ei saa olla erikoismerkkejä, rivinvaiho- tai kappaleenjakomerkkiä, joten näitä merkkeja ei saa käyttää tässä. Jos tiivistelmäsi ei sisällä erikoimerkkejä eikä kaipaa kappaleenjakoa, voit hyödynttää makroa abstracttext luodessasi lomakkeen tiivistelmää (katso kommentti tässä TeX-tiedostossa alla). Metadatatiivistelmatekstin on muuten oltava sama kuin lomakkeessa oleva teksti.

Avainsanat Vastus, resistanssi, lämpötila

Aalto-yliopisto
Sähkötekniikan
korkeakoulu

Författare Gabriel Gomes Ziegler

Titel Arbetets titel — Opinnäytteen mahdollinen alaotsikko

Utbildningsprogram Electronik och electroteknik

Huvudämne Sopiva pääaine

Övervakare Prof. Pirjo Professori

Handledare TkD Alan Advisor, DI Elsa Expert

Samarbetspartner Company or institute name in Swedish (if relevant)

Datum 21.9.2023 **Sidantal** 28+1 **Språk** engelska

Sammandrag

Sammandraget är en kort beskrivning av arbetets centrala innehåll: vad undersöktes, hur undersöktes det och vilka var de viktigaste resultaten?

I lärdomsprov som skrivs på svenska skrivs sammandraget på svenska och engelska, på motsvarande sätt skrivs sammandraget på finska och engelska i lärdomsprov på finska. Finsk- eller svenskspråkiga studerande som skriver sitt lärdomsprov på engelska ska skriva sammandraget på engelska och på sitt skolutbildningsspråk. Studerande vars skolutbildningsspråk inte är svenska eller finska skriver sammandraget endast på engelska. Den studerande kan om hen så önskar lägga till ett andra eller tredje sammandrag på sitt eget modersmål. Sammandraget fungerar då ofta som mognadsprov och bör i så fall vara minst 300 ord långt. Information om mognadsprov på svenska finns på MyCourses:
https://mycourses.aalto.fi/course/view.php?id=26872.

Nyckelord Nyckelord på svenska, temperatur

Preface

Thanks notes

Otaniemi, 31 August 2024

Eddie E. Engineer

Contents

DL	Deep Learning	15
NLP	Natural Language Processing	11
PDF	Portable Document Format	12
OCR	Optical Character Recognition	12
LLM	Large Language Model	7
GPT	Generative Pre-trained Transformer	7
BERT	Bidirectional Encoder Representations from Transformers	10
KPI	Key Performance Indicator	12
RAG	Retrieval Augmented Generation	7
ESG	Environmental, Social, and Governance	17
GHG	Greenhouse Gas	17
JSON	JavaScript Object Notation	18
MAPE	Mean Absolute Percentage Error	7
DPI	Dots Per Inch	23
RLHF	Reinforcement Learning from Human Feedback	13

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 LLMs, such as 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

Ever since Portable Document Format (PDF)s were created by Adobe in 1993 [1], they have been used to store and share information. These document standard quickly became a way of companies reporting their financial information for the public as well as Key Performance Indicator (KPI)s and other important information internally. This has led to a large amount of information being stored in PDFs, which has led to a need to extract information from these files. A series of professions have arised from this need, such as data entry, data extraction, and data analysis. The extraction of information from PDFs has been a manual process for most of the tasks until recent years, when Optical Character Recognition (OCR) and NLP technologies have been developed to automate processes involving processing PDFs.

The procedure of extracting information from a document — recently referred to as "Document AI" [2] — 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 text and visual form. This is particularly true for financial reports, where information is presented in text, tables, charts, and infographics. The problem is further complicated by the fact that financial reports are often not standardized, and the information is presented in diverse range of formats.

2.1 LLMs

Large Language Models (LLMs) are a class of artificial intelligence models that have been designed to understand, generate, and interact with human language at a large scale. These models are trained on vast amounts of text data, allowing them to learn language patterns, grammar, context, and even domain-specific knowledge. As a result, LLMs can perform a wide range of language-related tasks, such as translation, summarization, question answering, and more, with remarkable proficiency. The development and evolution of LLMs have been instrumental in advancing the field of natural language processing (NLP), enabling more natural and effective human-computer interactions. The capabilities of LLMs have found applications in various sectors, including but not limited to customer service, content creation, and, notably, in extracting and analyzing information from documents in the field known as Document AI [?].

2.2 GPT

2.3 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 [3]. 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 [3]. 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 [3].

2.4 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 [4].

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 models like GPT 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 [5]. 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 [6]. 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 [7], fine-tuning a model on a specific domain [8] or by retrying the generation process multiple times while validating the output against a defined model such as the tools *LangChain* [9] and *instructor* [10] propose. In this study, we show applications of how using defined models with *pydantic* [11] and *instructor* can help in mitigating hallucinations in the context of document data extraction.

2.7.2 Interpretability and Explainability

Ever since Deep Learning (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. 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 [12, 13]. 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

Dataset used to benchmark different methods.

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 Environmental, Social, and Governance (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:

- **Greenhouse Gas (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.

- **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.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \tag{1}$$

- **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.

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

4.4.2 MAPE

For numerical errors, the 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.

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100 \tag{3}$$

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 LLM 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:

```
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 [14].

- 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 [15].*

- 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 pre-processed 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

5.1 Limitations of the dataset

5.2 Limitations of the data extraction systems

Explain what are the observed limitations

6 Conclusions

References

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

[2] 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

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

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

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

[6] 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

[7] 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

[8] 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.

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

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

[11] 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/

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

[13] 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

[14] 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.

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

A Contents of an appendix